

# THWARTING SELFISH AND MALICIOUS BEHAVIOR IN WIRELESS NETWORKS

THÈSE N° 3449 (2006)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

MARIO ČAGALJ

Ingenieur Electricien Diplômé (B.Sc.), University of Split, Croatie  
de nationalité croate

acceptée sur proposition du jury:

Prof. Jean-Pierre Hubaux, directeur de thèse

Prof. Emre Telatar, président de jury

Prof. Karl Aberer, rapporteur

Prof. Virgil D. Gligor, rapporteur

Prof. Edward W. Knightly, rapporteur

Dr. Philippe Oechslin, rapporteur

Lausanne, EPFL

2006



# Abstract

Security is at the core of any communication system and, in particular, of wireless (radio) networks. In this thesis, we focus on three important security aspects in the framework of wireless networks: *selfish (noncooperative) behavior* at the Medium Access Control (MAC) layer, “*radio channel jamming*”-based *Denial-of-Service (DoS) attacks* against sensor networks and *secure key agreement* in peer-to-peer wireless networks.

In the context of selfish behavior at the MAC layer, we focus on single collision domain Carrier-Sense Multiple-Access with Collision Avoidance (CSMA/CA) networks. We use both cooperative and non-cooperative game theory to model and analyze the co-existence of multiple CSMA/CA selfish users. Using insights from the game theoretic analysis, we propose a simple channel access protocol that discourages selfish behavior and results in the optimal and fair allocation of the available bandwidth. We perform an extensive evaluation of the proposed protocol.

We then consider two types of malicious behavior. The first type deals with an adversary who tries to obstruct the operation of a wireless network by jamming the used radio channel. The second type is concerned with an adversary who interferes with a key agreement protocol executed between parties that use a radio link, in an attempt to learn their private information or to fool them into accepting fake messages as genuine.

Concerning the first kind of malicious behavior, we focus on wireless sensor networks, perhaps the most vulnerable category of wireless networks to this kind of threat. An adversary can mask the events that the sensor network should detect by stealthily jamming an appropriate subset of the nodes; in this way, he prevents them from reporting what they sense to the network operator. Therefore, in spite of the fact that an event is sensed by one or several nodes (and the sensor network is fully connected), the network operator cannot be informed on time – we call this the *coverage paradox*. To mitigate this problem, we propose a reactive defense mechanism based on *wormholes*, which were so far considered to be a security threat. In our solution, thanks to channel diversity, the nodes under the jamming attack are able to create (probabilistically) a communication route that is resistant to jamming; thus, appropriate information can be conveyed out of the jammed region. We develop appropriate mathematical models to study the proposed mechanisms.

Concerning the second kind of malicious behavior, we focus on the problem of a *user-friendly* key agreement (and message authentication) in settings where the users do not share any authenticated secret or certified public key in advance. We base our approach on the Diffie-Hellman key agreement protocol, which is known to be vulnerable to the “man-in-the-middle” attack if the users involved in the protocol do not share any authenticated information about each other (e.g., public keys, certificates, passwords, shared keys, etc.) prior to the protocol execution. We solve the problem by leveraging on the natural ability of users to authenticate each other by visual and verbal contact. We propose three techniques: the first is based on the visual comparison of short strings, the second on distance bounding, and the third on a novel concept called *integrity codes (I-codes)*. In each case, the users do not need to enter any password or other data, nor do they need physical or infrared

connectivity between their devices. We analyze our protocols using a well-established methodology that leads us to a rigorous modularization and a thorough robustness proof of our proposal. We also provide an implementation of *I*-codes.

# Résumé

La sécurité est au cœur de tout système de communication et en particulier des réseaux radio sans-fil. Dans cette thèse de doctorat, nous considérons trois aspects importants de sécurité qui concernent les réseaux sans-fils: L'attitude égoïste et non-coopérative au niveau de la couche MAC (Medium Access Control), les attaques DoS (Denial-of-Service) des réseaux de senseurs basées sur le brouillage du canal de communication radio, et l'établissement sécurisé de clés dans les réseaux point-à-point sans fil.

Dans le contexte d'une attitude égoïste au niveau de la couche MAC, nous nous concentrons sur les réseaux CSMA/CA (Carrier-Sense Multiple-Access with Collision Avoidance) à domaine de collision unique. Nous utilisons la théorie des jeux coopératifs et non-coopératifs pour modéliser et analyser la coexistence de plusieurs utilisateurs CSMA/CA égoïstes. En se basant sur l'analyse des théories des jeux, nous proposons un protocole simple d'accès au canal de communication. Ce protocole décourage l'attitude égoïste et a pour résultat une répartition optimale et équitable de la bande passante. Nous évaluons le protocole proposé de façon approfondie.

Ensuite, nous considérons deux types de comportement malicieux. Le premier type concerne un attaquant qui essaye de compromettre le bon fonctionnement du réseau sans fil en brouillant le canal de communication radio. Le deuxième type concerne un attaquant qui interfère dans un protocole de partage de clefs exécuté par deux parties utilisant le lien radio; l'attaquant a pour but de recouvrer les données privées des deux parties et de leur faire accepter de faux messages comme étant authentiques.

En ce qui concerne le premier type de comportement malicieux, nous considérons les réseaux de senseurs sans fil qui représente peut-être la catégorie de réseaux sans fil la plus vulnérable à ce genre d'attaque. Un attaquant peut masquer les événements que le réseau de senseurs devrait détecter en brouillant imperceptiblement un sous-ensemble judicieusement choisi de nœuds; ainsi, il les empêche de rapporter les événements qu'ils détectent à l'opérateur du réseau. Par conséquent, l'opérateur peut ne pas être informé à temps d'un événement qui pourtant à été détecté par un ou plusieurs nœuds, même si le réseau de senseurs est totalement connecté; nous appelons ceci le *coverage paradox*. Pour atténuer l'effet de ce problème, nous proposons un mécanisme de défense réactif basé sur les *wormholes*, qui ont été jusqu'ici considérés comme une menace pour la sécurité. Dans notre solution, grâce à la l'existence de différents canaux de communication, les nœuds qui subissent le brouillage peuvent créer (avec une certaine probabilité) une voie de communication qui ne soit pas brouillée; ainsi, l'information à propos des événements peut être transmise en dehors de la région brouillée. Nous développons les modèles mathématiques appropriés pour étudier les mécanismes proposés.

Pour ce qui est du deuxième type de comportement malicieux, nous considérons le problème d'établissement de clés (et d'authentification de messages) *user-friendly* dans le cas où les utilisateurs ne partagent à l'avance aucune donnée secrète authentifiée, ni aucune clef publique certifiée. Nous basons notre approche sur le protocole Diffie-Hellman d'échange de clés, qui est connu pour

être vulnérable à l'attaque *man-in-the-middle* si les deux utilisateurs impliqués dans le protocole ne partagent aucune information authentifiée (par exemple, des clefs publiques, des certificats, des mots de passe, des clefs partagées, etc...) avant l'exécution de protocole. Nous résolvons le problème en exploitant la capacité naturelle des utilisateurs à s'authentifier par le contact visuel et verbal. Nous proposons trois techniques : la première est basée sur la comparaison visuelle de petites chaînes de caractère, la seconde est basée sur la technique du *distance bounding*, et la troisième est basée sur un nouveau concept appelé codes d'intégrité (*I-codes*). Dans chacun de ces cas, les utilisateurs n'ont besoin d'aucune connectivité physique ou infrarouge entre leurs machines, pas plus que de mot de passes ni d'aucune autre donnée. Nous analysons nos protocoles en utilisant une méthodologie bien établie qui nous mène à une modularisation rigoureuse et à une preuve exhaustive de la robustesse de notre proposition. Nous fournissons également une implémentation des *I-codes*.

# Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Jean-Pierre Hubaux who has given me the opportunity to “grow up” in a scientific sense and to fully develop my ideas. Jean-Pierre was very supportive during my PhD journey and helped me whenever I was in need.

I would also like to thank the members of my thesis jury Prof. Emre Telatar, Prof. Karl Aberer, Prof. Virgil D. Gligor, Prof. Edward W. Knightly and Dr. Philippe Oechslin for their effort in reading and judging this work.

I am grateful to former and present members of LCA1 for all their support and encouragement, as well as for the great time I had working in this group; Thanks to Prof. Srdjan Čapkun, Naouel Ben Salem, Mark Felegyhazi, Dr. Imad Aad, Jacques Panchard, Maxim Raya, Jun Luo and Prof. Levente Buttyán. I am especially thankful to my office-mate Naouel for her help with numerous French puzzles (including Résumé). I would also like to thank other LCA students and professors for creating a very stimulating working environment and for many enlightening discussions: Prof. Patrik Thiran, Prof. Jean-Yves LeBoudec, Prof. Mathias Grossglauser, Dr. Sonja Buchegger, Dr. Božidar Radunović, Dr. Ljubica Blažević, Dr. Olivier Dousse, Mathilde Durvy, Alaeddine El Fawal, Ruben Merz, Hung Nguyen, Gianluca Rizzo, Slaviša and Nataša Sarafijanović, Dr. Milan Vojnović, Dominique Tschopp, Maciej Kurant, Michal Piorkowski, Dr. Daniel Figueiredo and Dr. Emre Koksal.

I want to thank LCA technical and administrative staff for their help and support during my thesis, Jean-Pierre Dupertuis, Marc-André Luthi, Philippe Chammartin, Danielle Alvarez, Holly Cogliati, and Angela Devenoge.

I would like to thank my friends from Lausanne for their support and help: Srdjan Čapkun and Elizabeta Čavar, Milan and Sandra Vojnović, Gorana Čapkun and Markus Niggli.

I am especially thankful to my friends from Split for being true friends, Igor and Nataša Marinović, Željko and Ana Lovrinčević.

I am indebted to my mother Anka, my father Mate, my sister Ivana, my brother Marko, my grandmas Iva and Mara, my brother in law Mario and my niece Barbara, for all their love, encouragement and support. I would also like to thank my mother in law Smiljana, my father in law Josip and my sister in law Gorana for their support and encouragement.

Finally, a very special thanks to my son Goran and my wife Marijana for their unconditional love, support, encouragement and patience. It is only with you, Goran and Mary. that I can do anything.

The work presented in this thesis was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (<http://www.terminodes.org>). I thankfully acknowledge this support.





# Contents

Abstract	iii
Résumé	v
Acknowledgements	vii
<b>Introduction</b>	<b>1</b>
<b>1 Selfish Behavior in CSMA/CA Networks</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Game Theory . . . . .	4
1.2.1 Noncooperative Games . . . . .	4
1.2.2 Nash Bargaining Framework and Cooperative Games . . . . .	6
1.2.3 Essential Games and Robust Equilibria . . . . .	8
1.3 System Model and Assumptions . . . . .	8
1.4 Single Stage Noncooperative Game . . . . .	10
1.4.1 Characterization of Utility Functions $u_i(W)$ . . . . .	10
1.5 Nash Equilibria of the CSMA/CA Game . . . . .	13
1.6 Uniqueness, Fairness and Pareto Optimality . . . . .	16
1.7 Multiple Stage CSMA/CA Game . . . . .	21
1.7.1 Nash Equilibria of the Repeated Game . . . . .	21
1.7.2 Practical Penalty Function . . . . .	23
1.7.3 Equilibrium Coordination Algorithms . . . . .	25
1.8 Implementation . . . . .	27
1.8.1 Detection Mechanism . . . . .	27
1.8.2 Adaptive Strategy . . . . .	28
1.8.3 Reaching the Pareto-optimal Point . . . . .	30
1.9 Related Work . . . . .	32
1.10 Summary . . . . .	33
<b>2 Wormhole Defense: New Anti-Jamming Techniques in Sensor Networks</b>	<b>35</b>
2.1 Introduction . . . . .	35
2.2 Motivation and Existing Tradeoffs . . . . .	36
2.2.1 Proactive vs. Reactive Sensor Networks . . . . .	37
2.2.2 Straightforward Solutions Might Not Be Adequate . . . . .	37
2.3 Proposed Solution: Probabilistic Wormholes . . . . .	38
2.4 Wormholes via Wired Pairs of Sensor Nodes . . . . .	38

2.4.1	Rationale of Wired Pairs . . . . .	38
2.4.2	Performance Analysis . . . . .	40
2.4.3	Simulations and Model Validation . . . . .	45
2.5	Wormholes via Coordinated Frequency Hopping Pairs . . . . .	47
2.5.1	Rationale of Frequency Hopping (FH) Pairs . . . . .	47
2.5.2	Analysis of the FH Pairs Based Solution . . . . .	49
2.5.3	Simulations and Model Validation . . . . .	52
2.6	Wormholes via Uncoordinated Channel-Hopping . . . . .	53
2.6.1	Rationale of the Approach . . . . .	53
2.6.2	System Model and Assumptions . . . . .	54
2.6.3	Performance with an Inactive Attacker . . . . .	55
2.6.4	Performance with an Active Attacker . . . . .	58
2.6.5	Simulations . . . . .	60
2.7	Related Work . . . . .	62
2.8	Summary . . . . .	63
<b>3</b>	<b>Key Agreement in Peer-to-Peer Wireless Networks</b>	<b>65</b>
3.1	Introduction . . . . .	65
3.2	Problem Statement and Assumptions . . . . .	66
3.2.1	Threats Against Radio-Based Systems . . . . .	66
3.2.2	Assumptions . . . . .	67
3.2.3	Commitment Schemes . . . . .	68
3.3	Message Transfer (MT) Authenticator . . . . .	68
3.3.1	Straightforward Approaches are Suboptimal or Flawed . . . . .	68
3.3.2	Optimal MT-Authenticator Based on String Comparison . . . . .	69
3.4	From Secure MT-Authenticator to Secure Key Agreement . . . . .	72
3.4.1	Straightforward Application of the MT-SC Authenticator . . . . .	73
3.4.2	Diffie-Hellman Key Agreement with String Comparison (DH-SC) . . . . .	74
3.5	Security Analysis of the MT-Authenticator . . . . .	76
3.5.1	Matching Conversations . . . . .	76
3.5.2	Security of the MT-SC Authenticator . . . . .	77
3.6	Diffie-Hellman Key Agreement Based on Distance Bounding (DH-DB) . . . . .	79
3.6.1	Properties of DH-DB Protocol . . . . .	79
3.6.2	Implementation . . . . .	81
3.7	Related Work . . . . .	82
3.8	Summary . . . . .	83
<b>4</b>	<b>Integrity (<i>I</i>) codes for Message Integrity Protection Over Insecure Channels</b>	<b>85</b>
4.1	Introduction . . . . .	85
4.2	Problem Statement and Assumptions . . . . .	86
4.3	Integrity ( <i>I</i> )-codes . . . . .	88
4.3.1	Definition . . . . .	88
4.3.2	<i>I</i> -codes on a Radio Channel . . . . .	89
4.3.3	Preventing the Attacker from Erasing Symbol “1” . . . . .	90
4.3.4	Synchronization and Complementary Encoding . . . . .	92
4.4	Implementation . . . . .	95
4.5	Authentication Through Presence . . . . .	97

4.5.1	Access Point Authentication . . . . .	98
4.5.2	Key Establishment over Insecure Channels . . . . .	99
4.6	Security Analysis of <i>I</i> -codes . . . . .	100
4.6.1	Anti-Blocking Property of the Radio Channel . . . . .	101
4.6.2	Randomization at the Sender: the Impact of Spreading . . . . .	103
4.6.3	Energy Content of the Emitted Signals . . . . .	105
4.7	Related Work . . . . .	107
4.8	Summary . . . . .	107
<b>Conclusion</b>		<b>109</b>
<b>Appendices</b>		<b>111</b>
<b>A</b>	<b>Minimum-Energy Broadcasting in All-Wireless Networks</b>	<b>111</b>
A.1	Introduction . . . . .	111
A.2	System Model . . . . .	112
A.3	Complexity Issues . . . . .	112
A.3.1	General Graph Version . . . . .	113
A.3.2	Geometric Version . . . . .	115
A.4	Proposed Algorithms . . . . .	116
A.4.1	$O(\log N)$ -approximation Algorithm . . . . .	116
A.4.2	A Heuristic Based Approach . . . . .	119
A.4.3	Distributed Implementation of EWMA . . . . .	121
A.5	Performance Evaluation . . . . .	124
A.6	Related Work . . . . .	126
A.7	Summary . . . . .	127
<b>Bibliography</b>		<b>128</b>
<b>Index</b>		<b>136</b>



# Introduction

As the popularity of mobile devices such as PDAs, laptops, and mobile phones increases every day, users tend to rely on them in a growing number of situations. Due to the rapid increase in the number of users of wireless communication services, it becomes difficult to provide centralized solutions to both known and emerging security threats. This is even more true given that wireless networks are intrinsically more vulnerable to different kinds of *malicious* behaviors, such as eavesdropping and Denial-of-Service (DoS) attacks. Furthermore, in a growing number of situations, users' communications take place in unlicensed frequency bands (for example, using IEEE 802.11a/b/g or Bluetooth) that quickly become saturated. As a result, some users will be tempted to increase their share of the available bandwidth by manipulating their network adapters. Such a (*selfish*) behavior is clearly undesirable, since it can lead to an inefficient usage of the available bandwidth and ultimately to frequent collapses of wireless networks.

Traditionally, such security challenges are solved by means of a centralized authority. However, today, as users can easily set up their own networks and are highly mobile, this paradigm is not appropriate as it does not scale well and it usually comes at a high monetary cost. Therefore, a new approach to the design of protocols should be taken. Of course, this shift in design paradigm must not affect the security or efficiency of the developed protocols and communication mechanisms.

This thesis is concerned with several security issues in this new self-organized setting that lacks any centralized authority, more specifically:

- how to efficiently arbitrate channel access between multiple selfish users, in a self-organized way (Chapter 1),
- how to ensure timely data delivery in the presence of an attacker who jams a wireless sensor network, while relying on low-cost defense mechanisms (Chapter 2),
- how to provide *user-friendly* data integrity, authentication and secure key agreement services in self-organized settings where users cannot (or do not want to) rely on the centralized authority (Chapter 3 and Chapter 4)

Next, we outline the thesis. In Chapter 1, we use a game-theoretic approach to investigate the problem of selfish behavior of nodes in CSMA/CA networks, specifically geared towards the most widely accepted protocol in this class of protocols, IEEE 802.11. We identified two families of *Nash equilibria* [40] of the static CSMA/CA game. The first family is characterized by the classical *tragedy of the commons* result, that is, each selfish user receives zero throughput. The second family comprises Nash equilibria with the property that only one selfish user receives positive payoff (i.e., throughput), while the others get zero. We further show that no Nash equilibrium of the single stage CSMA/CA game is robust to infinitesimally small changes in the payoff functions. Therefore, contrary to the intuition, the tragedy of the commons is *not* a robust outcome of the CSMA/CA game!

We then use the Nash bargaining framework [39] (from cooperative game theory) to identify a desirable solution of the CSMA/CA game that exhibits the following three properties: (i) the solution is unique, (ii) the solution results in a fair distribution of the system throughput, and (iii) the solution results in system optimum allocation of the available capacity. Finally, we formulate a dynamic game where the solution obtained from the Nash bargaining framework is a unique (and robust) Nash equilibrium point.

In Chapter 2, we show that *wormholes* [49], which were so far considered to be a threat, can be used as a reactive defense mechanism against radio jamming attacks. We explain the principle of *probabilistic wormholes* by analyzing three approaches based on this principle. In the first, a network with regular wireless sensor nodes is augmented with a certain number of wired pairs of sensor nodes, therefore resulting in a *hybrid sensor network*. In the second, the deployed nodes (or a subset of them) organize themselves as frequency hopping pairs. For both approaches we compute the probability that at least one wormhole can be formed. Finally, in the third approach, there is no coordination about either the communication channel or communication slots; we analyze this approach through simulations.

In Chapter 3, we propose two approaches to the problem of *user-friendly* key agreement (and mutual authentication) in settings where the users do not share any authenticated information in advance. The first approach belongs to the family of solutions requiring the users to compare strings of words, whereas the other approach is completely novel; it is based on radio-channel specific techniques, namely, *distance-bounding*. In addition, we make the following contributions: (i) we design protocols that are provably secure in a realistic communication model, (ii) we apply a modular approach to designing and analyzing the protocols, thus paving the way to the design of *re-usable* (provably secure) message transfer (MT) authenticators, and (iii) we significantly increase user-friendliness compared to existing approaches.

In Chapter 4, we propose *integrity codes (I-codes)*, a coding scheme that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (i.e. public keys or shared secret keys). The construction of *I-codes* enables a sender to encode any message such that if its integrity is violated in transmission, the receiver is able to detect it. We analyze in detail the use of *I-codes* on a radio communication channel and we present their implementation on Mica2 wireless sensor platform as a *proof of concept*. We finally show how *I-codes* can be used for several applications, including for key establishment and for broadcast authentication over an insecure radio channel. We perform a detailed analysis of the security of our coding scheme and we show that it is secure within a realistic attacker model.

At the beginning of my PhD track, we also worked on the problem of constructing *minimum-energy broadcasting trees* in static wireless networks. In Appendix A, we report on some relevant results we achieved in this context. There, we focus on the problem of power-optimal broadcast, for which it is well known that the broadcast nature of the radio transmission can be exploited to optimize energy consumption. We provide a formal proof of NP-completeness for the general case and give an NP-completeness result for the geometric case; in the former, the network topology is represented by a generic graph with arbitrary weights, whereas in the latter a Euclidean distance is considered. For the general case, we show that it cannot be approximated better than  $O(\log N)$ , where  $N$  is the total number of nodes. We then describe an approximation algorithm that achieves the  $O(\log N)$  approximation ratio. We also describe a new heuristic, Embedded Wireless Multicast Advantage. We show that it compares well with other proposals and we explain how it can be distributed.

# Chapter 1

## Selfish Behavior in CSMA/CA Networks

### 1.1 Introduction

Carrier-sense multiple-access with collision avoidance (CSMA/CA) protocols rely on the random deferment of packet transmissions for the efficient use of a shared wireless channel among many nodes in a network; this class of MAC protocols is one of the most popular for wireless networks. In general, it is assumed that all nodes respect the rules of the protocol. We believe, however, that this assumption is less and less appropriate, because the network adapters are becoming more and more *programmable* [90]. As a result, today a user can modify the behavior of his wireless interface very easily. In this chapter, we study the stability and efficiency of wireless networks that contain one or several *selfish* users. By “selfish” we designate the users who are ready to tamper with their wireless interface in order to increase their own share of the common transmission resource; we assume these users to be rational, and not malicious (they are willing to harm other users only if they can derive a benefit from this misbehavior).

More specifically, we consider that a selfish user (*cheater*) makes use of the easiest (and yet highly rewarding) cheating technique, specifically he deliberately does not respect the random deferment of the transmission of his packets (see Figure 1.1). Although this cheating technique is straightforward, we show that studying its implications is far from trivial. In order to better understand possible outcomes of selfish behavior on the MAC layer, we make use of both *noncooperative* and *cooperative* game theory. In our analysis, each node (a selfish user) is a player, the throughput it enjoys is its payoff, and the size of its contention window represents its move. By making use of this model and of extensive simulations, we systematically study several problems. First, we consider the simple case of a network with a single cheater. We then assume the presence of several cheaters, and identify two families of *Nash equilibria* in a *single stage* (i.e., *static*) game: one family always results in a network collapse, and in the other, there is only one selfish user who receives non-null throughput. We also show that the equilibria resulting in the network collapse (*the tragedy of the commons-equilibria*) are not *robust* (to arbitrarily small perturbations of the users’ payoff functions).

Since the Nash equilibria of the static game are either highly unfair or highly inefficient, we look for an alternative solution. In this regard, we compute the fair Pareto-optimal point of an operation of such a system. We then show how to make the Pareto-optimal point a Nash equilibrium point by using the theory of *dynamic (repeated) games*. We introduce the notion of *cooperative players*, specifically cheaters who try to continue operating at the fair Pareto-optimal point of operation. We

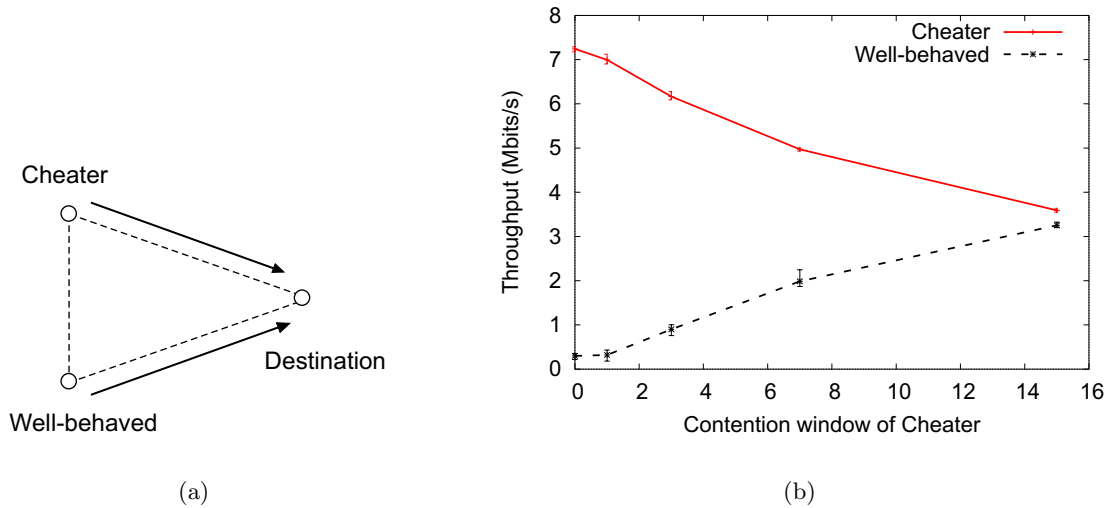


Figure 1.1: Using a configurable IEEE 802.11b wireless card, (e.g., Atheros), a cheater can reduce his contention window size to increase his share of throughput at the expense of the well-behaved node: (a) Experimented topology – one cheater and one well-behaved node send full data rate UDP flows to the common destination (dashed lines represent the connectivity and the flows are represented by solid arcs); (b) Resulting throughputs as a function of the cheater’s contention window size – the bars show max and min values of 5 real tests (the nominal data rate is 11 Mb/s).

also propose a detection and a punishment technique for those players who exhibit a noncooperative behavior. Finally, we explain how the players can collectively search for the Pareto-optimal point of operation, even if they are unaware of the number of nodes present in the network.

The organization of this chapter is the following. In Section 1.2, we give a short introduction to the theory of noncooperative and cooperative games. In Section 1.3, we introduce our system model and give related assumptions. In Section 1.4, we apply noncooperative game theory to analyze Nash equilibria of the single stage CSMA/CA game. In Section 1.6, we use cooperative game theory (the Nash Bargaining Framework) to reason about the optimal capacity allocation. In Section 1.7, we apply the theory of repeated games to show how the optimal allocation from the previous section can be supported as a Subgame Perfect Nash Equilibrium. In Section 1.8, using the insights from Section 1.7, we implement a distributed algorithm that leads the network nodes to the Pareto optimal Subgame Perfect Nash equilibrium. We address the related work in Section 1.9. Finally, we summarize the chapter in Section 1.10.

## 1.2 Game Theory

In this section, we introduce some definitions and the terminology from game theory, which we use in our analysis.

### 1.2.1 Noncooperative Games

The theory of noncooperative games studies the behavior of selfish players in any situation where each player’s optimal choice may depend on his forecast of the choices of his opponent. The word



“noncooperative” means that the players’ choices are based only on their perceived self-interest and that they do not try to find an agreement with the other players [40].

In this thesis, we will consider noncooperative games in *strategic* (or *normal*) form [40]. A game in normal form has three elements: the set of players  $\mathcal{I} = \{1, 2, \dots, I\}$ ,  $I < \infty$ , the *pure strategy set*  $S_i$  for each player  $i$ , and the *payoff functions*  $u_i$  that give player  $i$ ’s utility  $u_i(s)$  for each strategy profile  $s = (s_1, \dots, s_I)$ . We will denote all players other than some given player  $i$  by “ $-i$ ” (e.g.,  $u_i(s_i, s_{-i}) \equiv u_i(s_1, \dots, s_i, \dots, s_I) \equiv u_i(s)$ ). We note here that the normal form can model not only those noncooperative games in which the players act simultaneously and once for all, but also the extensive-form games, which include explicitly the timing of the players’ decisions.

A *mixed strategy* is a probability distribution over pure strategies. Each player  $i$  chooses a probability distribution over his set of pure strategies  $S_i$  (independently of probability distributions of his opponents). We denote with  $\sigma_i(s_i)$  the probability that the distribution  $\sigma_i$  assigns to  $s_i \in S_i$ ; we assume the set  $S_i$  to be finite for all players  $i \in \mathcal{I}$ . The payoffs to a profile of mixed strategies  $\sigma = \{\sigma_1, \dots, \sigma_I\}$  are the expected values of the corresponding pure-strategy payoffs. Player  $i$ ’s payoff  $u_i(\sigma) \equiv u_i(\sigma_i, \sigma_{-i})$  to profile  $\sigma$  is

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{s \in S} \left( \prod_{j=1}^I \sigma_j(s_j) \right) u_i(s), \quad (1.1)$$

where  $S \equiv \times_{i \in \mathcal{I}} S_i$ . It is interesting to observe that the following holds

$$\begin{aligned} u_i(\sigma_i, \sigma_{-i}) &= \sum_{s_i \in S_i} \sigma_i(s_i) \left\{ \sum_{s_{-i} \in S_{-i}} u_i(s_i, s_{-i}) \left( \prod_{j \neq i} \sigma_j(s_j) \right) \right\} \\ &= \sum_{s_i \in S_i} \sigma_i(s_i) u_i(s_i, \sigma_{-i}) \\ &\leq \max_{s_i \in S_i} u_i(s_i, \sigma_{-i}), \end{aligned} \quad (1.2)$$

where by definition  $u_i(s_i, \sigma_{-i}) = \sum_{s_{-i} \in S_{-i}} u_i(s_i, s_{-i}) \left( \prod_{j \neq i} \sigma_j(s_j) \right)$ , and the last inequality follows from the fact that  $\sum_{s_i \in S_i} \sigma_i(s_i) = 1$ .

In other words, in the given game, no player can do better than playing his best response pure-strategies; conditioned on the fact that pure strategy equilibria exist in the considered game.

**Definition 1** We say that a pure strategy  $s_i$  is strictly dominated for player  $i$  if there exists a mixed strategy  $\sigma_i$  such that  $u_i(\sigma_i, s_{-i}) > u_i(s_i, s_{-i})$  for all  $s_{-i} \in S_{-i}$ .

The strategy  $s_i$  is *weakly dominated* if there exists a  $\sigma_i$  such that  $u_i(\sigma_i, s_{-i}) \geq u_i(s_i, s_{-i})$ , and there exists at least one  $s'_{-i}$  such that  $u_i(\sigma_i, s'_{-i}) > u_i(s_i, s'_{-i})$ .

**Definition 2** A mixed-strategy profile  $\sigma^*$  is a Nash equilibrium if, for all players  $i$ ,

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \text{ for all } s_i \in S_i.$$

A *pure-strategy Nash equilibrium* is a pure strategy profile  $s^*$  that satisfies the same conditions. In other words, at the Nash equilibrium  $s^*$  (or,  $\sigma^*$ ) no player has an incentive to change unilaterally his strategy  $s_i^*$  (or,  $\sigma_i^*$ ). That it suffices to check that no player has a profitable pure-strategy

deviation, follows from the inequality in (1.2) above. In order to study the existence of Nash equilibria, we use the notion of a player's *best-response function* (or *correspondence*) [84].

For any  $s_{-i} \in S_{-i}$  define  $B_i(s_{-i})$  to be the set of player  $i$ 's best actions given  $s_{-i}$ :

$$B_i(s_{-i}) = \{s_i \in S_i : u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) \text{ for all } s'_i \in S_i\} . \quad (1.3)$$

We call the set-valued function  $B_i$  the best-response function of player  $i$ . We can restate the definition of a Nash equilibrium as follows.

**Definition 3** *A pure-strategy profile  $s^*$  is a Nash equilibrium if and only if*

$$s_i^* \in B_i(s_{-i}^*) \text{ for all players } i \in \mathcal{I} .$$

The same obviously holds for a mixed-strategy profile  $\sigma^*$ . In this thesis, we will focus on pure-strategy Nash equilibria. The reason is that our noncooperative game admits pure-strategy Nash equilibria (as we show later in this chapter), and from the expression (1.2) we know that a player cannot do better than playing his best response pure-strategies.

## 1.2.2 Nash Bargaining Framework and Cooperative Games

Generally, selfish players can achieve better payoffs by cooperating with each other. Since the set of feasible payoffs (or outcomes) in a cooperative game is generally very large, the players should have a *means* to agree on a reasonable outcome, that is, the outcome that results in the distribution of the gains from cooperation in a manner that reflects properly the *bargaining power* of the different players. The *Nash Bargaining Framework (NBF)* provides the means for a set of players (with equal bargaining power) to negotiate on which point of the set of feasible payoffs they will agree upon [39].

It is convenient to derive a bargaining problem from the normal form of an  $I$ -players game  $G = \langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$  that we have introduced in Section 1.2.1. An *outcome* in the game  $G$  corresponds to a strategy profile  $s = (s_i)_{i \in \mathcal{I}}$ . The payoff function of player  $i$  is then defined as  $u_i : S \rightarrow \mathbb{R}$ , where  $S = \times_{i \in \mathcal{I}} S_i$  and  $\mathbb{R}$  is the set of real numbers. The set of all feasible payoffs is defined as

$$U = \{u : u = (u_1(s), \dots, u_I(s)), s \in S\} .$$

Note that  $U \subset \mathbb{R}^I$ . The Nash bargaining framework is used to model a situation in which the players from the set  $\mathcal{I}$  negotiate on which point from the set  $U \subset \mathbb{R}^I$  they will agree upon. An important element of the Nash bargaining framework is a *fixed disagreement point*  $u^\circ = (u_1^\circ, \dots, u_I^\circ) \in U$ . It is common to define  $u_i^\circ$  for all players  $i$  as follows

$$u_i^\circ = \min_{s_{-i} \in S_{-i}} \max_{s_i \in S_i} u_i(s_i, s_{-i}) . \quad (1.4)$$

The role of the disagreement point is to provide an incentive for the agreement point to take effect; in case the negotiations break down, the outcome becomes the strategy profile  $s^\circ$  resulting in the payoff profile  $u^\circ$ . Given a disagreement point  $u^\circ \in U$ , the pair  $B = (U, u^\circ)$  is called a *bargaining problem*.

We can derive another bargaining problem  $B = (C, c^\circ)$  by extending the set of feasible outcomes  $U$  to its convex hull  $C$ . Here we define  $c^\circ = u^\circ$ . Notice that any element  $c \in C$  can be represented as  $c = \sum_{k=1}^m \alpha_k u^{(k)}$ , ( $m \leq I + 1$ ), where  $u^{(k)} = (u_1(s^{(k)}), \dots, u_I(s^{(k)})) \in U$ , ( $s^{(k)} \in S$ ),  $\alpha_k \geq 0$ , and  $\sum_{k=1}^m \alpha_k = 1$ . Note that  $\alpha_k$  can be thought of as the probability of an outcome  $s^{(k)} \in S$  taking place;

a vector  $\alpha$  defines lotteries over deterministic outcomes  $s^{(k)} \in S$ . Therefore,  $c = (c_1, \dots, c_I) \in C$  is the expected payoff to the players.

To solve bargaining problem  $B = (C, c^\circ)$ , Nash took an axiomatic approach and proposed a one-point solution  $f(C, c^\circ) \in C$  to  $B$ . Let  $\mathcal{B}$  denote the set of all pairs  $(C, c^\circ)$  such that

- (i)  $C \subset \mathbb{R}^I$  is compact and convex
- (ii)  $\exists c \in C$  such that  $c > c^\circ$ .

A function  $f : \mathcal{B} \rightarrow \mathbb{R}^I$  is called the *Nash Bargaining Solution (NBS)* of the bargaining problem, if it satisfies the following properties.

**IUO** The bargaining solution  $f(\cdot)$  is *independent of utility origins (IUO)*, that is, for any  $\beta = (\beta_1, \dots, \beta_I) \in \mathbb{R}^I$  we have

$$f_i(C', c^\circ + \beta) = f_i(C, c^\circ) + \beta_i \text{ for every player } i, \quad (1.5)$$

whenever  $C' = \{(c_1 + \beta_1, \dots, c_I + \beta_I) : c \in C\}$ .

The *IUO* property says that the bargaining solution does not depend on absolute scales of utility.

**IUU** The bargaining solution  $f(\cdot)$  is *independent of utility units (IUU)*, that is, for any  $\beta = (\beta_1, \dots, \beta_I) \in \mathbb{R}^I$  we have

$$f_i(C', c^\circ) = \beta_i f_i(C, c^\circ) \text{ for every player } i, \quad (1.6)$$

whenever  $C' = \{(\beta_1 c_1, \dots, \beta_I c_I) : c \in C\}$ .

With the *IUO* property, the *IUU* property says that the bargaining solution does not involve interpersonal comparisons of utilities.

**P** The bargaining solution  $f(\cdot)$  satisfies the *Pareto* property (*P*), that is,

$$\nexists c = (c_1, \dots, c_I) \in C \text{ such that } c_i > f_i(C, c^\circ) \text{ for every player } i. \quad (1.7)$$

**S** The bargaining solution  $f(\cdot)$  satisfies the property of *symmetry (S)*, that is,

$$\text{if } C \subset \mathbb{R}^I \text{ is a symmetric set, then } c_i^* = c_j^*, \forall i, j \in \mathcal{I}, \quad (1.8)$$

where  $c^* = f(C, c^\circ)$ .

The *S* property says that if all players are identical, then the gains from cooperation are split equally.

**IAA** The bargaining solution  $f(\cdot)$  satisfies the property of *independence of irrelevant alternatives (IIA)*, that is,

$$\text{if } C' \subset C \text{ and } f(C, c^\circ) \in C', \text{ then } f(C', c^\circ) = f(C, c^\circ). \quad (1.9)$$

The *IIA* condition says that if  $f(C)$  is the outcome in  $C$  and we consider a  $C'$  that is smaller than  $C$  but retains the feasibility of  $f(C)$ , that is, we only eliminate from  $C$  irrelevant alternatives, then  $f(C)$  remains the outcome in  $C'$ .

**Theorem 1 (Nash Bargaining Solution [39])** *There is a unique function  $f(\cdot)$  satisfying properties IUO, IUU, P, S and IAA. Furthermore, for all bargaining problems  $(C, c^\circ) \in \mathcal{B}$ , the point  $f(C, c^\circ) = (c_1^*, \dots, c_I^*)$  is the unique solution of the maximization problem*

$$\begin{aligned} & \text{maximize} && \prod_{i=1}^I (c_i - c_i^\circ) \\ & \text{subject to} && c \in C \\ & && c \geq c^\circ . \end{aligned} \tag{1.10}$$

### 1.2.3 Essential Games and Robust Equilibria

In practice it is unlikely that the game modeler will have specified payoff functions that are perfectly correct. The issue coming out of this observation is whether equilibrium predictions of the original game with payoffs  $u$  are approximate equilibrium predictions of the real game with nearby payoffs  $\hat{u}$ . We now define the notion of *proximity* in finite games [40, Section 12.1.2]. Let

$$u = (u_i(s))_{i \in \mathcal{I}, s \in S}$$

and

$$\hat{u} = (\hat{u}_i(s))_{i \in \mathcal{I}, s \in S}$$

denote two payoff profiles, and let

$$\sigma = (\sigma_i(s_i))_{i \in \mathcal{I}, s_i \in S_i}$$

and

$$\hat{\sigma} = (\hat{\sigma}_i(s_i))_{i \in \mathcal{I}, s_i \in S_i}$$

denote two mixed strategy profiles. Let

$$D(u, \hat{u}) = \max_{i \in \mathcal{I}, s \in S} |u_i(s) - \hat{u}_i(s)| \tag{1.11}$$

and

$$d(\sigma, \hat{\sigma}) = \max_{i \in \mathcal{I}, s_i \in S_i} |\sigma_i(s_i) - \hat{\sigma}_i(s_i)| . \tag{1.12}$$

**Definition 4** *A Nash equilibrium  $\sigma$  of game  $u$  is essential or robust if for any  $\varepsilon > 0$  there exists  $\eta > 0$ , such that for any  $\hat{u}$  such that  $D(u, \hat{u}) < \eta$  there exists a Nash equilibrium  $\hat{\sigma}$  of game  $\hat{u}$  such that  $d(\sigma, \hat{\sigma}) < \varepsilon$ . A game  $u$  is essential if all its equilibrium points are essential.*

## 1.3 System Model and Assumptions

We consider  $N$  wireless nodes that are willing to transmit data to  $N$  designated receivers ( $N < \infty$ ). All the nodes use the same radio channel. We assume all the nodes to share the same collision domain, that is, each node can hear any other node (see Figure 1.2). This is to avoid complications introduced by the *hidden terminal problem*. Nodes use a CSMA/CA based protocol to resolve contention at the Medium Access Control (MAC) layer. In this chapter, we will be dealing exclusively with IEEE 802.11 (in the Distributed Coordination Function (DCF) mode) [30]; however, we note that the analysis carried out in this setting can also be extended to other CSMA/CA based protocols. We further assume each node to have an authentic MAC layer identifier (the MAC

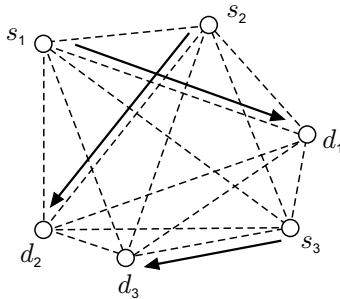


Figure 1.2: An example of a single-collision domain network with  $N = 3$  communicating pairs. Dashed lines represent the connectivity and the flows between the pairs are represented by solid arcs.

address). This can be achieved by means of MAC layer authentication. Finally, we assume that the nodes are static and that they always have packets (of the same size) to send.

We consider a scenario where out of the  $N$  senders, a subset  $\mathcal{I}$  of  $I$  sending nodes deliberately deviate from the IEEE 802.11 protocol. Without any loss of generality, we assume  $\mathcal{I} = \{1, \dots, I\}$ . We designate the nodes of subset  $\mathcal{I}$  as *cheaters*. There can be a number of ways in which a node can cheat. For example, in violation of the standard protocol, a cheater  $i \in \mathcal{I}$  initializes his contention window size to a lower value in order to obtain a higher throughput (cf. Figure 1.1). We will call this lower value  $W_i$ . Moreover, a cheater does not respect the *binary exponential backoff* [30] principle and keeps his contention window size fixed after a collision, i.e. equal to  $W_i$ . This mode of cheating is the easiest for potential cheaters, since it does not require changes to be made in the operation of the IEEE 802.11 protocol. We would like to stress that the main conclusions of this chapter are applicable to any other cheating technique. The relevance of these misbehaving techniques becomes even higher with the emerging standards that address the Quality of Service support, such as IEEE 802.11e [107]. The latter gives the users total control of the MAC parameters, therefore enabling them to easily cheat.

We assume the cheaters in our model to be *rational*, that is, they want to maximize their own benefit. In this particular context, every cheater  $i \in \mathcal{I}$  seeks to maximize the average throughput  $r_i$  he enjoys. This problem can easily be modelled in the game theoretic framework introduced in Section 1.2. The cheater nodes define the set  $\mathcal{I}$  of players in this game. We define the pure-strategy set  $S_i$  of a given player  $i$  as follows

$$S_i = \{1, 2, \dots, W_{max}, W_\infty\} , \quad (1.13)$$

where  $W_{max} < \infty$  is a positive integer and the symbol  $W_\infty$  means that the player  $i$  does not transmit, which is equivalent to  $W_i = \infty$ ; note that the set  $S_i$  is finite. The strategy of each player  $i$  consists in setting the value of his contention window  $W_i \in S_i$  such that player  $i$ 's payoff function  $u_i$  is maximized. Since we assume that each player  $i$  tries to maximize his own throughput, we define a player  $i$ 's utility function  $u_i$  to be equal to the enjoyed throughput  $r_i^{(c)}$ , that is,

$$u_i(W) = r_i^{(c)}(W) , \quad (1.14)$$

where  $W = (W_1, \dots, W_I, W_{I+1}, \dots, W_N)$ , and  $W_j \in S_j$ , ( $j \leq I$ ), are strategies chosen by players  $j \in \mathcal{I}$ , while contention windows  $W_k$ , ( $I + 1 \leq k \leq N$ ), belong to the *well-behaved* nodes. Here the

superscript “(c)” denotes a cheater. In our game theoretic analysis, we will often neglect the well-behaved nodes. So we will often use  $W$  to denote  $W = (W_1, \dots, W_I)$ . Finally, we denote the game as defined above by  $G_{\text{CSMA/CA}} = \langle N, \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (r_i^{(c)})_{i \in \mathcal{I}} \rangle$  and call it the *CSMA/CA game*.

## 1.4 Single Stage Noncooperative Game

We first analyze the problem of misbehaving from the perspective of a single cheater and then consider more complex scenarios with multiple cheaters in the system. In this section, we consider the interaction of multiple cheaters in a *single stage* of the CSMA/CA game  $G_{\text{CSMA/CA}}$ , where the players choose their strategies only once and keep playing them forever.

### 1.4.1 Characterization of Utility Functions $u_i(W)$

In order to characterize the payoff functions  $u_i(W) = r_i^{(c)}(W)$ , ( $i \in \mathcal{I}$ ), we first have to understand the relationship between the contention window profiles  $W = (W_1, \dots, W_I, \dots, W_N)$  and the resulting payoffs  $r_i^{(c)}(W)$ . For this purpose, we make use of the celebrated Bianchi’s model for the saturation throughput of the IEEE 802.11 protocol [19]. Since we assume that a cheater’s objective is to maximize his throughput (and we assume he always has a packet to send), he will tend to use the full channel capacity (i.e., the system will operate at the saturation point). Therefore, we make use of the same model as [19].

To estimate the throughput of IEEE 802.11, in a network with no misbehaving nodes, Bianchi [19] used a two-dimensional Markov chain of  $m$  backoff stages in which each stage represents the backoff time counter of a node. A transition takes place upon collision and successful transmission, to a higher stage and to the first stage respectively. Considering the stationary distribution of the chain, the *channel access probability*  $\tau$  of a node is derived as a function of the number of backoff stage levels  $m$  and the minimum contention window value  $W_{\min}$ :

$$\tau = \frac{2}{1 + W_{\min} + pW_{\min} \sum_{k=0}^{m-1} (2p)^k} \quad (1.15)$$

where  $p$  is the conditional probability that a transmitted packet collides, that is:

$$p = 1 - (1 - \tau)^{N-1} \quad (1.16)$$

where  $N$  is the number of the contending nodes. Equations (1.15) and (1.16) form a system of two nonlinear equations that has a unique solution [19].

The throughput enjoyed by a given node  $i$ , which is the average information payload transmitted in a slot time over the average length of a slot time, can be computed using Bianchi’s model as follows:

$$r_i = \frac{P_i^s L}{P^s T^s + P^c T^c + P^{id} T^{id}} \quad (1.17)$$

where  $P_i^s = \tau_i \prod_{j \neq i} (1 - \tau_j)$  is the probability that the station  $i$  successfully transmits during a random time slot ( $j \neq i$  is a shorthand notation for  $j \in \{1, \dots, N\} \setminus \{i\}$ );  $L$  is the average packet payload size;  $P^s = \sum_{j=1}^N P_j^s$ ;  $T^s$  is the average time needed to transmit a packet of size  $L$  (including the inter-frame spacing periods [19]);  $P^{id} = \prod_{j=1}^N (1 - \tau_j)$  is the probability of the channel being idle;  $T^{id}$  is the duration of the idle period (a single slot);  $P^c = 1 - P^{id} - \sum_{j=1}^N P_j^s$  is the probability

of collision; and  $T^c$  is the average time spent in the collision. Note that we must have the following satisfied  $P^s + P^c + P^{id} = 1$ .

To describe a network with cheating nodes we use two separate Markov chains; we note that both Markov chains are studied in [19]. The first, with  $m = 0$  (no exponential backoff, since cheaters are assumed to fix their contention windows (Section 1.3)), is used to derive the channel access probabilities  $\tau_i^{(c)}$  of cheaters  $i \in \mathcal{I}$ . The second chain, with  $m > 0$  [19], is used to derive the access probabilities  $\tau_j^{(w)}$  of well-behaved (non cheating) nodes. The conditional collision probabilities are derived considering both well-behaved and cheating nodes access probabilities.

Since cheater  $i$  does not respect the backoff procedure of IEEE 802.11 (i.e.,  $m = 0$ ), his channel access probability degenerates to

$$\tau_i^{(c)} = \frac{2}{W_i + 1}, \quad (1.18)$$

where  $W_i$  is the cheater  $i$ 's contention window size [19]. The channel access probability for well-behaved nodes,  $\tau_j^{(w)}$ , is expressed by

$$\tau_j^{(w)} = \frac{2}{1 + W_{min} + p^{(w)} W_{min} \sum_{k=0}^{m-1} (2p^{(w)})^k} \quad (1.19)$$

where

$$p^{(w)} = 1 - \left(1 - \tau_j^{(w)}\right)^{N-I-1} \prod_{i \in \mathcal{I}} \left(1 - \tau_i^{(c)}\right), \quad (1.20)$$

Note that expression (1.20) is the generalization of expression (1.16) in the presence of cheaters. Note also that  $\tau_j^{(w)}$  is the same for all the well-behaved nodes and so we set  $\tau_j^{(w)} = \tau^{(w)}$ . After a straightforward algebraic manipulation of equation (1.17), we obtain the following expression for the throughput  $r_i^{(c)}$  of a cheater  $i$ :

$$r_i^{(c)} = \frac{\tau_i^{(c)} c_i^{(1)}}{\tau_i^{(c)} c_i^{(2)} + c_i^{(3)}}, \quad (1.21)$$

where

$$c_i^{(1)} = p_{-i} L \quad (1.22)$$

$$c_i^{(2)} = p_{-i} (T^s - T^{id}) - s_{-i} (T^s - T^c) \quad (1.23)$$

$$c_i^{(3)} = (1 - p_{-i} - s_{-i}) T^c + s_{-i} T^s + p_{-i} T^{id}, \quad (1.24)$$

where we have used the following substitutions

$$\begin{aligned} p_{-i} &= \prod_{j \in \mathcal{I} \setminus \{i\}} \left(1 - \tau_j^{(c)}\right) \left(1 - \tau^{(w)}\right)^{N-I} \\ s_{-i} &= \sum_{j \in \mathcal{I} \setminus \{i\}} \tau_j^{(c)} \prod_{k \in \mathcal{I} \setminus \{i, j\}} \left(1 - \tau_k^{(c)}\right) \left(1 - \tau^{(w)}\right)^{N-I}. \end{aligned} \quad (1.25)$$

Note here, that the only parameter that a cheating node  $i$  has a control over is its own  $W_i$ . By varying  $W_i$ , a node changes its own access probability  $\tau_i^{(c)} = f(W_i)$ , as well as the access probability  $\tau^{(w)} = f(W)$ , ( $W = (W_1, \dots, W_i, \dots, W_N)$ ,  $W_i = W_{min}$ ,  $i = \{I + 1, \dots, N\}$ ), of the well-behaved nodes; this follows from expressions (1.18), (1.19) and (1.20).

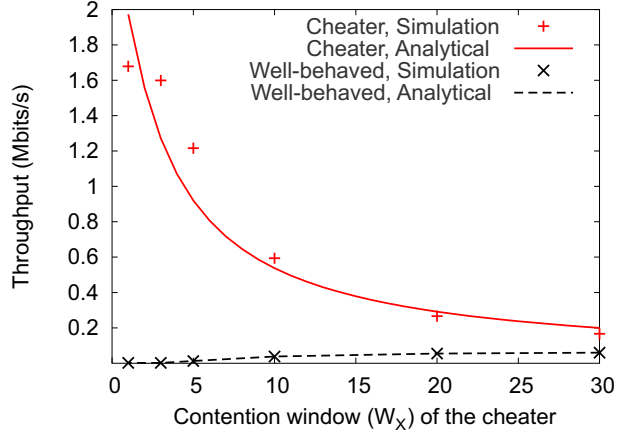


Figure 1.3: Throughputs for  $N = 20$  nodes, out of which one is a cheater.

For mathematical convenience, let us assume for the moment that  $W_i$  for every cheater  $i \in \mathcal{I}$  is a continuous variable. Although the access probabilities of the well-behaved nodes (and thus the expressions  $c_i^{(1)}$ ,  $c_i^{(2)}$  and  $c_i^{(3)}$ ) depend on  $\tau_i^{(c)}$ , we neglect this dependence for a first degree analysis. This approximation allows us to elaborate a closed form expression of the first derivative of equation (1.21):

$$\frac{\partial r_i^{(c)}}{\partial W_i} = \frac{\partial r_i^{(c)}}{\partial \tau_i^{(c)}} \frac{\partial \tau_i^{(c)}}{\partial W_i} = \frac{c_i^{(1)} c_i^{(3)}}{\left(\tau_i^{(c)} c_i^{(2)} + c_i^{(3)}\right)^2} \frac{-2}{(W_i + 1)^2} \leq 0. \quad (1.26)$$

If  $\tau_j^{(c)} < 1$  for all  $j \in \mathcal{I} \setminus \{i\}$ , then we have a strict inequality in (1.26). Therefore, as expected, the received throughput  $r_i^{(c)}$  is a strictly decreasing function of  $W_i$  (for  $\tau_j^{(c)} < 1, \forall j \in \mathcal{I} \setminus \{i\}$ ). Thus, by unilaterally decreasing its own  $W_i$ , a selfish node can increase its received throughput (except if  $\tau_j^{(c)} = 1$ , for some cheater  $j \neq i$  – as we will see in the following section, this case has important implications on the set of Nash equilibria of the CSMA/CA game). We stress here that this conclusion would remain the same even if we considered the dependence of  $c_i^{(1)}$ ,  $c_i^{(2)}$  and  $c_i^{(3)}$  on  $\tau_i^{(c)}$ . In fact, by using this approximation, we actually underestimate the benefits of the cheater (the cheater gets more throughput in reality).

We will now verify this claim (and the modified Bianchi's model) by simulations performed in *ns-2* [3]. The simulation setup<sup>1</sup>, summarized in Table 1.1, consists of  $N = 20$  sender nodes. A node  $X$  deliberately fails to adhere to the protocol and tries to misbehave following the cheating model presented in Section 1.3. The parameter values for the IEEE 802.11 protocol are chosen according to the IEEE 802.11b standard [30]. The duration for each simulation run is 50 seconds and the results are averaged over 5 simulation runs.

Figure 1.3 plots the throughput obtained by cheater  $X$ , as well as by each well-behaved node for different values of  $W_X$ . Simulation results show a good match with the analytical results. As can be observed from Figure 1.3, the throughput obtained by the cheater increases monotonically with the decrease in  $W_X$ .

<sup>1</sup>In the rest of the chapter, we will only mention the changes that are done from this reference simulation setup.



Table 1.1: Simulation parameters

Parameter	Value
Topology	100 m $\times$ 100 m, random
Receive range	240 m
Propagation	Free space
MAC	802.11b
Scheme	Basic (No RTS/CTS)
Channel capacity	2 Mbits/s
Traffic sources	CBR / UDP, 1050-byte frames each 5 ms

Now that we have characterized the cheaters' payoff functions  $u_i(W) = r_i^{(c)}(W)$ , we next study Nash equilibria of the single stage game  $G_{\text{CSMA/CA}}$ .

## 1.5 Nash Equilibria of the CSMA/CA Game

In this section we do not consider well-behaved nodes (i.e., we assume  $N = I$ ). We will focus only on pure-strategy Nash equilibria (Section 1.2), since, as we will show soon, they exist in  $G_{\text{CSMA/CA}}$  and we know from the expression (1.2) in Section 1.2 that no player can do better than playing his best response pure-strategies.

We will study the existence of Nash equilibria by making use of the concept of a player's best-response function introduced in Section 1.2.1. Let us introduce the following notations

$$\begin{aligned} W_{-i} &= (W_1, \dots, W_{i-1}, W_{i+1}, \dots, W_I) \\ S_{-i} &= \{S_1, \dots, S_{i-1}, S_{i+1}, \dots, S_I\}, \end{aligned}$$

where  $S_i$  are the pure-strategy sets of the players (cf. expression (1.13)). Following the exposition in Section 1.2.1, we define a player  $i$ 's best-response (set valued) function  $B_i(W_{-i})$  as follows

$$B_i(W_{-i}) = \left\{ W_i \in S_i : r_i^{(c)}(W_i, W_{-i}) \geq r_i^{(c)}(W'_i, W_{-i}) \text{ for all } W'_i \in S_i \right\}.$$

Then from Definition 3 (Section 1.2.1), we know that a pure-strategy profile  $W^* = (W_1^*, \dots, W_I^*)$  is a Nash equilibrium if and only if  $W_i^* \in B_i(W_{-i}^*)$  for every player  $i \in \mathcal{I}$ .

**Lemma 1** *For any strategy profile  $W$  that constitutes a Nash equilibrium in  $G_{\text{CSMA/CA}}$ ,  $\exists i \in \mathcal{I}$  such that  $W_i = 1$ .*

*Proof:* Assume by contradiction that  $W = (W_1, \dots, W_I)$  is a Nash equilibrium such that  $W_k > 1$ ,  $\forall k \in \mathcal{I}$ . Now, take one player, say  $i$ , and consider his best-response function  $B_i(W_{-i})$ . Since  $r_i^{(c)}$  is a strictly decreasing function of  $W_i$  (equation (1.26) and  $W_k > 1 \Rightarrow \tau_k^{(c)} < 1$ ,  $\forall k \in \mathcal{I}$  (equation (1.18))), it follows readily that the only value of  $W_i$  that satisfies

$$r_i^{(c)}(W_i, W_{-i}) \geq r_i^{(c)}(W'_i, W_{-i}) \text{ for all } W'_i \in S_i,$$

is unity, that is,  $B_i(W_{-i}) = \{1\}$ . Since, by definition, at any Nash equilibrium  $W_i \in B_i(W_{-i})$ , we have  $W_i = 1$ . However, this contradicts our initial assumption that  $W_i > 1$ , which concludes the proof.  $\square$

**Theorem 2** *The game  $G_{\text{CSMA/CA}}$  admits exactly  $(W_{\max} + 1)^I - W_{\max}^I$  Nash equilibria.*

*Proof:* Assume that for some player  $i \in \mathcal{I}$  we have  $W_i = 1$ . Then his access probability  $\tau_i^{(c)} = 1$  and consequently for all players  $k \in \mathcal{I} \setminus \{i\}$  it follows that  $r_k^{(c)} = 0$  for any value of  $W_k \in S_k$  (equation (1.21)). Therefore, for any value of  $W_k \in S_k$  we have  $W_k \in B_k(W_{-k})$ , where  $k \in \mathcal{I} \setminus \{i\}$ . This clearly holds for any number of players who have their contention window set to unity. Combining this with Lemma 1, we obtain the following characterization of Nash equilibria:

**(Nash equilibria)** At any Nash equilibrium of  $G_{\text{CSMA/CA}}$  we have at least one cheater who sets his contention window to unity and all the other cheaters play any strategy from  $\{1, \dots, W_{\max}, W_{\infty}\}$ .

Finally, the theorem follows by observing that out of the total of  $(W_{\max} + 1)^I$  different strategy profiles  $W = (W_1, \dots, W_I)$ , ( $W_j \in \{1, \dots, W_{\max}, W_{\infty}\}$ ), exactly  $W_{\max}^I$  do not contain any unity element.  $\square$

It is interesting to observe that the equilibria can be classified in two families. To describe these, we define a set  $\mathcal{D} = \{i : W_i = 1, i \in \mathcal{I}\}$ .

**1st family:**  $|\mathcal{D}| = 1$ , that is, there is only one player  $i \in \mathcal{I}$  who plays  $W_i = 1$  and receives a non-null throughput  $r_i^{(c)} > 0$ , and  $r_k = 0$  for all players  $k \in \mathcal{I} \setminus \{i\}$ .

**2nd family:**  $|\mathcal{D}| > 1$ , that is, we have more than one player  $i \in \mathcal{I}$  who play strategy  $W_i = 1$ , in which case  $r_k^{(c)} = 0$ , for all players  $k \in \mathcal{I}$ .

Note that some Nash equilibria from the first family are also Pareto optimal. For example, a strategy profile  $W = (1, W_2 = W_{\infty}, \dots, W_I = W_{\infty})$  is a Pareto optimal Nash equilibrium, since players  $\mathcal{I} \setminus \{i\}$  do not actually transmit (i.e.,  $W_i = W_{\infty} = \infty \Rightarrow \tau_i^{(c)} = 0$ ) and player 1 gets all the system capacity for himself. This is interesting, since it is seldom the case that Nash equilibria are also Pareto optimal. The equilibria from the second family are known as *the tragedy of the commons* in economics.

It is important to emphasize that in our search for Nash equilibria, we take into account weakly dominated strategies (see Definition 1). If, on the contrary, we do not consider weakly dominated strategies (a usual practice), the only Nash equilibrium of the game  $G_{\text{CSMA/CA}}$  is the strategy profile  $W = (W_i = 1)_{i \in \mathcal{I}}$ . It is generally believed that for noncooperative channel access games (as the one studied in this chapter) the strategy profile  $W = (W_i = 1)_{i \in \mathcal{I}}$  is the unique and robust Nash equilibrium. However, we have shown here (Theorem 2) that it is certainly not the only Nash equilibrium of  $G_{\text{CSMA/CA}}$ . We next show that this strategy profile is neither robust in the game  $G_{\text{CSMA/CA}}$ .

Let us define an approximate game  $\hat{G}_{\text{CSMA/CA}}$  to our original game  $G_{\text{CSMA/CA}}$  as follows

$$\hat{G}_{\text{CSMA/CA}} = \langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (\hat{u}_i)_{i \in \mathcal{I}} \rangle, \text{ with } \hat{u}_i(W) = r_i^{(c)}(W) - \begin{cases} \epsilon_i, & \text{if } W_i < W_{\infty}; \\ 0, & \text{if } W_i = W_{\infty} \end{cases}, \forall i \in \mathcal{I},$$

where  $\epsilon_i$  is an infinitesimally small but positive constant (i.e.,  $0 < \epsilon_i \ll 1$ ) that satisfies the following:  $r_i^{(c)}(W) > \epsilon_i, \forall W$  such that  $r_i^{(c)}(W) > 0$ ; the existence of such a constant follows from the fact that the number of nodes in the system is finite ( $N < \infty$ ).

Intuitively, the *cost* term  $\epsilon_i$  says that a player prefers not transmitting at all than transmitting unsuccessfully. Being infinitesimally small, the cost term  $\epsilon_i$  does not change significantly the player  $i$ 's payoff function  $u_i$ . Let us now look at the equilibria of the game  $\hat{G}_{\text{CSMA/CA}}$ .

**Theorem 3** *A strategy profile  $W$  is a Nash equilibrium of the game  $\hat{G}_{\text{CSMA/CA}}$  if and only if*

$$\exists! i \in \mathcal{I} \text{ such that } W_i = 1 \text{ and } W_j = W_\infty, \forall j \in \mathcal{I} \setminus \{i\} .$$

*Proof:* It is easily seen that Lemma 1 applies to game  $\hat{G}_{\text{CSMA/CA}}$  too. Now, consider again the case where for some player  $i \in \mathcal{I}$  we have  $W_i = 1$ . Then his access probability  $\tau_i^{(c)} = 1$  and consequently for all players  $k \in \mathcal{I} \setminus \{i\}$  it follows that  $r_k^{(c)} = 0$  for any value of  $W_k \in S_k$  (equation (1.21)). This further implies  $u_k(W) = -\epsilon_k \leq 0$ . The best response function for player  $k$  in game  $\hat{G}_{\text{CSMA/CA}}$  is

$$\hat{B}_k(W_{-k}) = \left\{ W_k \in S_k : \hat{u}_k(W_k, W_{-k}) \geq \hat{u}_k(W'_k, W_{-k}) \text{ for all } W'_k \in S_k \right\} .$$

Then,  $\hat{B}_k(W_{-k}) = \{W_\infty\}, \forall k \in \mathcal{I} \setminus \{i\}$ , since  $u_k(W_k = W_\infty, W_{-k}) = 0 \geq -\epsilon_k$ . Also,  $\hat{B}_i((W_k = W_\infty)_{k \in \mathcal{I} \setminus \{i\}}) = \{1\}$  (Lemma 1). Therefore, a strategy profile

$$W = \left( W_i = 1, (W_k = W_\infty)_{k \in \mathcal{I} \setminus \{i\}} \right)$$

is a Nash equilibrium by Definition 3 (Section 1.2). We conclude the proof by observing that this is valid for an arbitrary player  $i \in \mathcal{I}$ .  $\square$

Therefore, by an infinitesimally small change in the original game's payoff functions, we have arrived at a game with a significantly different set of Nash equilibria: the set of Nash equilibria of  $\hat{G}_{\text{CSMA/CA}}$  is a small subset of those of  $G_{\text{CSMA/CA}}$ . Actually, all the Nash equilibria in  $\hat{G}_{\text{CSMA/CA}}$  are Pareto optimal and the strategy profile  $(W_i = 1)_{i \in \mathcal{I}}$  is not even an equilibrium point in  $\hat{G}_{\text{CSMA/CA}}$ . We conclude our study of robustness of the Nash equilibria of our original game  $G_{\text{CSMA/CA}}$  with the following theorem.

**Theorem 4** *The Nash equilibrium  $W = (W_i = 1)_{i \in \mathcal{I}}$  of the CSMA/CA game  $G_{\text{CSMA/CA}}$  is nonessential (it is not robust), and therefore the CSMA/CA game  $G_{\text{CSMA/CA}}$  is nonessential.*

*Proof:* The notion of essential games was introduced in Section 1.2.3. Observe first that the Nash equilibrium  $W = (W_i = 1)_{i \in \mathcal{I}}$  of  $G_{\text{CSMA/CA}}$  implies  $u_i(W) = 0, \sigma_i(W_i = 1) = 1$  and  $\sigma_i(W_i \in S_i \setminus \{1\}) = 0, \forall i \in \mathcal{I}$ ; recall  $\sigma_i(W_i)$  is the probability that player  $i$  assigns to strategy  $W_i \in S_i$ . Let us also fix the following equilibrium of game  $\hat{G}_{\text{CSMA/CA}}$

$$\hat{W} = (\hat{W}_1 = W_\infty, \hat{W}_2 = 1, \hat{W}_3 = W_\infty, \dots, \hat{W}_I = W_\infty) .$$

Note that this implies  $\hat{\sigma}_i(\hat{W}_i = 1) = 0, \forall i \in \mathcal{I} \setminus \{2\}$ .

To prove this theorem, we next calculate the distances  $D(\cdot)$  and  $d(\cdot)$  between the payoff vectors  $u$  and  $\hat{u}$ , and between the strategy vectors  $\sigma$  and  $\hat{\sigma}$  of the games  $G_{\text{CSMA/CA}}$  and  $\hat{G}_{\text{CSMA/CA}}$ , respectively.

Using the definitions introduced in Section 1.2.3, we have

$$\begin{aligned} D(u, \hat{u}) &= \max_{i \in \mathcal{I}, W \in \times_{i \in \mathcal{I}} S_i} |u_i(W) - \hat{u}_i(W)| \\ &\leq \max_{i \in \mathcal{I}, W \in \times_{i \in \mathcal{I}} S_i} \epsilon_i \\ &= \eta, \end{aligned}$$

where  $\eta > 0$  is an infinitesimally small but positive value; this follows from the definition of  $\epsilon_i$ . Similarly, for the distance between strategy profiles we have

$$\begin{aligned} d(\sigma, \hat{\sigma}) &= \max_{i \in \mathcal{I}, W_i \in S_i} |\sigma_i(W_i) - \hat{\sigma}_i(W_i)| \\ &\stackrel{(1)}{\geq} \max_{i \in \mathcal{I}, W=(W_i=1)_{i \in \mathcal{I}}} |\sigma_i(W_i) - \hat{\sigma}_i(W_i)| \\ &= \max_{i \in \mathcal{I}} |\sigma_i(W_i = 1) - \hat{\sigma}_i(W_i = 1)| \\ &\stackrel{(2)}{=} 1, \end{aligned}$$

where the inequality (1) follows from the fact that we reduce the maximization domain and the equality (2) follows from the two fixed Nash equilibria  $W$  and  $\hat{W}$ . But then it follows immediately from Definition 4 (Section 1.2.3) that the Nash equilibrium  $W$  is not essential (robust) and consequently the game  $G_{\text{CSMA/CA}}$  is nonessential.  $\square$

We conclude that, contrary to the intuition (and somewhat common belief in the networking community), the tragedy of the common equilibria of the game  $G_{\text{CSMA/CA}}$  are not robust.

## 1.6 Uniqueness, Fairness and Pareto Optimality

We saw in the earlier section that, generally, there exist two families of Nash equilibria in the CSMA/CA game  $G_{\text{CSMA/CA}}$ . In the first family, we have great unfairness (a single player gets some positive payoff). Recall that some of the equilibria from the first family are system (Pareto) optimal. In the second family, we have highly inefficient equilibria resulting in a zero payoff for every player. Moreover, some equilibria of the game  $G_{\text{CSMA/CA}}$  are not robust, even to infinitesimally small payoff perturbations. Therefore, we look for an alternative solution to  $G_{\text{CSMA/CA}}$  by allowing the players to agree on the strategies they will use.

A *desirable solution* of the CSMA/CA game should exhibit the following three properties.

**(Uniqueness)** The solution should be unique. This is to avoid uncertainties with respect to what solution each player should choose.

**(Fairness)** The solution should result in a fair distribution of the system throughput.

**(Pareto optimality)** The solution should result in a Pareto optimal allocation of the available bandwidth.

In order to derive such a solution we use the Nash Bargaining Framework (NBF) introduced in Section 1.2.2. We know from that section that the NBF is used to model a situation in which the players negotiate on which point of the set of joint feasible payoffs  $R$  they will agree upon. As in

the previous section, we will focus our game theoretic analysis on the cheating nodes. In the case of the game  $G_{\text{CSMA/CA}}$ , the set of joint feasible payoffs is given as follows

$$R = \left\{ r^{(c)} = \left( r_1^{(c)}, \dots, r_I^{(c)} \right) : r_i^{(c)} = g_i(W), i \in \mathcal{I}, W \in S \right\}, \quad (1.27)$$

where the functions  $g_i(\cdot)$  are derived from expressions (1.18) and (1.21), and  $S = \times_{i \in \mathcal{I}} S_i$ . The important element in the Nash bargaining framework is a fixed disagreement vector  $r^\circ = (r_1^\circ, \dots, r_I^\circ)$  (see Section 1.2.2). For our problem, it is reasonable to define for every player  $i \in \mathcal{I}$

$$r_i^\circ = \min_{W_{-i} \in S_{-i}} \max_{W_i \in S_i} r_i^{(c)}(W_i, W_{-i}) = 0.$$

Hence, the disagreement point  $r^\circ$  becomes

$$r^\circ = (r_i^\circ = 0)_{i \in \mathcal{I}},$$

which implies that the corresponding strategy profile  $W^\circ$  is such that at least two or more players  $i$  play strategy  $W_i = 1$ . It is important to stress at this point that such a strategy profile is a Nash equilibrium of the single stage game  $G_{\text{CSMA/CA}}$  (see the characterization of Nash equilibria in the proof of Theorem 2). This gives a high credibility to the disagreement point  $r^\circ$ . Thus the whole bargaining problem, in the context of the game  $G_{\text{CSMA/CA}}$ , can be conveniently described by the pair  $(R, r^\circ)$ .

We know from Section 1.2.2 that a sufficient condition for the bargaining problem  $B$  to have a unique solution (the Nash Bargaining Solution) satisfying the Nash axioms *IUO*, *IUU*, *P*, *S* and *IAA* is that the set of joint payoffs is convex and compact (and there exists at least one feasible point strictly preferable to the disagreement point). However, the set of joint payoffs  $R$  in the case of the CSMA/CA game  $G_{\text{CSMA/CA}}$  is neither compact nor convex: it consists of a countably finite number of points  $r^{(c)}$ . Nevertheless, we will show that the bargaining problem  $(R, r^\circ)$ , with  $R$  being the non-convex and non-compact set of feasible payoffs in  $G_{\text{CSMA/CA}}$  has a unique solution satisfying all the Nash axioms (in particular Pareto-optimality (axiom *P*) and fairness (axiom *S*)). Therefore, the desirable solution defined at the beginning of this section is actually the Nash Bargaining Solution (NBS) of the problem  $(R, r^\circ)$ .

From the Nash bargaining framework, we know that the NBS is obtained as the unique solution of the maximization problem (1.10). Let us define the corresponding maximization problem for the bargaining problem  $(R, r^\circ)$  as follows

$$\begin{aligned} & \text{maximize} && \prod_{i \in \mathcal{I}} \left( r_i^{(c)} - r_i^\circ \right) \\ & \text{subject to} && r^{(c)} \in R \\ & && r^{(c)} \geq r^\circ. \end{aligned} \quad (1.28)$$

By taking the logarithm of the objective function of (1.28) and using the fact  $r_i^\circ = 0, \forall i \in \mathcal{I}$ , we obtain the equivalent maximization problem  $\mathbf{\Pi}_1$  [39]

$$\begin{aligned} & \text{maximize} && \sum_{i \in \mathcal{I}} \log \left\{ r_i^{(c)}(W) \right\} \\ & \text{subject to} && r^{(c)} = g(W) \\ & && r^{(c)} \geq 0 \\ & && W \in S, \end{aligned} \quad (1.29)$$

where  $g(W) \stackrel{def}{=} (g_1(W), \dots, g_I(W))$ . In order to solve the problem  $\mathbf{\Pi}_1$  we relax the integrality constraints  $W \in S$  by making  $W_i$  ( $\forall i \in \mathcal{I}$ ) continuous variables that take the values from the set  $[0, \infty]$ . For mathematical convenience we will be working with the access probabilities  $\tau_i \in [0, 1]$ ,  $i \in \mathcal{I}$ , instead of the size of the contention windows. With these changes we define the following optimization problem  $\mathbf{\Pi}_2$

$$\begin{aligned} & \text{maximize} && \sum_{i \in \mathcal{I}} \log \left\{ r_i^{(c)}(\tau) \right\} \\ & \text{subject to} && 0 \leq \tau \leq 1 . \end{aligned} \quad (1.30)$$

We observe that the objective function of the problem  $\mathbf{\Pi}_2$  is strictly concave and its domain is the convex set  $[0, 1]^I$ . Therefore,  $\mathbf{\Pi}_2$  admits a unique solution [22]. Let us denote the value of this solution with  $v_2$ . Since the problem  $\mathbf{\Pi}_2$  is a relaxed version of the problem  $\mathbf{\Pi}_1$ , we know that  $v_2 \geq v_1$ , where  $v_1$  is the value of the optimal solution of the problem  $\mathbf{\Pi}_1$ . To solve  $\mathbf{\Pi}_2$  we define the corresponding Lagrangian  $\mathcal{L}$  as follows

$$\mathcal{L} \left( r^{(c)}, \lambda \right) = \sum_{i \in \mathcal{I}} \log \left( r_i^{(c)} \right) - \sum_{i \in \mathcal{I}} \lambda_i (\tau_i - 1) . \quad (1.31)$$

It is known from the convex optimization theory that the Karush-Kuhn-Tucker (KKT) first order conditions are sufficient conditions for a concave function to be maximized over a convex set (assuming that the constraint qualification holds) [22]. Therefore, in the case of the problem  $\mathbf{\Pi}_2$  the KKT first-order conditions are sufficient optimality conditions. From the first-order KKT conditions we have

$$\frac{\partial \mathcal{L}}{\partial \tau_k} = \sum_{i \in \mathcal{I}} \frac{1}{r_i^{(c)}} \frac{\partial r_i^{(c)}}{\partial \tau_k} - \lambda_k = 0, \quad \forall k \in \mathcal{I} , \quad (1.32)$$

and

$$\lambda_k (\tau_k - 1) = 0, \quad \forall k \in \mathcal{I} . \quad (1.33)$$

Now, it is easily seen that  $\lambda_k = 0, \forall k \in \mathcal{I}$ . This follows from the fact that there exists a feasible vector  $\tau < 1$  such that the optimal value of the equivalent to  $\mathbf{\Pi}_2$ , i.e.,  $\prod_{k \in \mathcal{I}} r_k^{(c)}$  is strictly positive; in case  $\tau_k = 1, \forall k \in \mathcal{I}$ , we have  $\prod_{k \in \mathcal{I}} r_k^{(c)} = 0$ . Therefore, conditions (1.32) change to

$$\frac{\partial \mathcal{L}}{\partial \tau_k} = \sum_{i \in \mathcal{I}} \frac{1}{r_i^{(c)}} \frac{\partial r_i^{(c)}}{\partial \tau_k} = 0, \quad \forall k \in \mathcal{I} , \quad (1.34)$$

Inspired by the symmetry property of the Nash Bargaining Solution, we next prove that the unique solution  $\tau^* = (\tau_1^*, \dots, \tau_I^*)$  of the problem  $\mathbf{\Pi}_2$  satisfies  $\tau_i^* = \tau_j^*, \forall i, j \in \mathcal{I}$ . We prove this by using the  $I$  KKT conditions (1.34). Since the constraints in the problem  $\mathbf{\Pi}_2$  satisfy the constraint qualification, the conditions (1.34) make a system of  $I$  independent equations with  $I$  unknowns  $\tau_i, i = \{1, \dots, I\}$ . Therefore, this system admits a unique solution. We show that there exists a value  $\tau^* \in (0, 1)$  such that  $\tau_i = \tau^*, \forall i \in \mathcal{I}$ , solves the system (1.34).

We first evaluate the partial derivatives  $\partial r_i^{(c)} / \partial \tau_k$  for all  $i \in \mathcal{I}$  and for the fixed  $k \in \mathcal{I}$  by making use of (1.21).

$$\frac{\partial r_k^{(c)}}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left( \frac{\tau_k c_k^{(1)}}{\tau_k c_k^{(2)} + c_k^{(3)}} \right) = \frac{p_{-k} L c_k^{(3)}}{\left( \tau_k c_k^{(2)} + c_k^{(3)} \right)^2} , \quad (1.35)$$

where  $p_{-i} = \prod_{j \in \mathcal{I} \setminus \{i\}} (1 - \tau_j)$ . Similarly,

$$\begin{aligned} \frac{\partial r_i^{(c)}}{\partial \tau_k} &= \frac{\partial}{\partial \tau_k} \left( \frac{\tau_i c_i^{(1)}}{\tau_i c_i^{(2)} + c_i^{(3)}} \right) = \frac{\partial}{\partial \tau_k} \left( \frac{\tau_i c_i^{(1)}}{\tau_k c_k^{(2)} + c_k^{(3)}} \right) \\ &= - \frac{\tau_i p_{-i,k} L}{\left( \tau_k c_k^{(2)} + c_k^{(3)} \right)^2} \left( c_k^{(2)} + c_k^{(3)} \right), \end{aligned} \quad (1.36)$$

where  $p_{-i,k} = \prod_{j \in \mathcal{I} \setminus \{i,k\}} (1 - \tau_j)$ . We postulate that  $\tau_i = \tau_j = \tau$ ,  $\forall i, j \in \mathcal{I}$ , and therefore we next evaluate the partial derivatives (1.35) and (1.36) at  $\tau$ ; for this reason, we will drop indexes from expressions  $c_i^{(2)}$  and  $c_i^{(3)}$  (given by (1.23) and (1.24), respectively).

$$\left. \frac{\partial r_k^{(c)}}{\partial \tau_k} \right|_{\substack{\tau_i = \tau \\ \forall i \in \mathcal{I}}} = (1 - \tau)^{I-1} L \frac{c^{(3)}(\tau)}{c(\tau)}, \quad (1.37)$$

where  $c(\tau) = \tau c^{(2)}(\tau) + c^{(3)}(\tau)$ . Similarly,

$$\left. \frac{\partial r_i^{(c)}}{\partial \tau_k} \right|_{\substack{\tau_i = \tau \\ \forall i \in \mathcal{I}}} = -\tau (1 - \tau)^{I-2} L \frac{c^{(2)}(\tau) + c^{(3)}(\tau)}{c(\tau)}. \quad (1.38)$$

By applying the same to the equations (1.34), they simplify to

$$\sum_{i \in \mathcal{I}} \left. \frac{\partial r_i^{(c)}}{\partial \tau_k} \right|_{\substack{\tau_i = \tau \\ \forall i \in \mathcal{I}}} = 0, \quad \forall k \in \mathcal{I}, \quad (1.39)$$

since  $r_i^{(c)}(\tau) = r_j^{(c)}(\tau)$ ,  $\forall i, j \in \mathcal{I}$ . Finally, by plugging the expressions (1.37) and (1.38) into the equations (1.39), they simplify to

$$F(\tau) \stackrel{def.}{=} \frac{\tau}{1 - \tau} \left( 1 + \frac{c^{(2)}(\tau)}{c^{(3)}(\tau)} \right) = \frac{1}{I - 1}, \quad \forall k \in \mathcal{I}, \quad (1.40)$$

where

$$\begin{aligned} c^{(2)}(\tau) &= (1 - \tau)^{I-1} (T^s - T^{id}) - (I - 1) \tau (1 - \tau)^{I-2} (T^s - T^c) \\ c^{(3)}(\tau) &= T^c - (1 - \tau)^{I-1} (T^c - T^{id}) + (I - 1) \tau (1 - \tau)^{I-2} (T^s - T^c). \end{aligned} \quad (1.41)$$

Note that  $c^{(3)}(\tau) > 0$ ,  $\forall \tau \in [0, 1]$  (since,  $T^s > T^c$ ). Furthermore, since  $0 \leq c^{(2)}(\tau) < \infty$  we have  $F(0) = 0$  and  $F(1) = \infty$ . Now, from  $1/(I - 1) \leq 1$  (for  $I \geq 2$ ) it follows that there exist at least one value of  $\tau$  such that the function  $y = F(\tau)$  intersects the line  $y = 1/(I - 1) \leq 1$ . Since  $F(0) = 0$ ,  $F(1) = \infty$  and  $0 < 1/(I - 1) \leq 1$ ,  $y = F(\tau)$  intersects  $y = 1/(I - 1)$  for values of  $\tau \in (0, 1)$ . Let us denote any such value by  $\tau^*$ . But then any point  $(\tau_1 = \tau^*, \dots, \tau_I = \tau^*)$  is a solution of the problem  $\mathbf{\Pi}_2$ . We already argued that the problem  $\mathbf{\Pi}_2$  admits a unique solution that solves the set of equations (1.39), that is, the equations (1.40). Therefore,  $\tau^*$  must be unique (i.e.,  $y = F(\tau)$  intersects  $y = 1/(I - 1)$  in a single point). We have thus proved the following theorem.

**Theorem 5** *The problem  $\mathbf{\Pi}_2$  admits a unique solution  $(\tau_1 = \tau^*, \dots, \tau_I = \tau^*)$ , with  $\tau^* \in (0, 1)$ .*

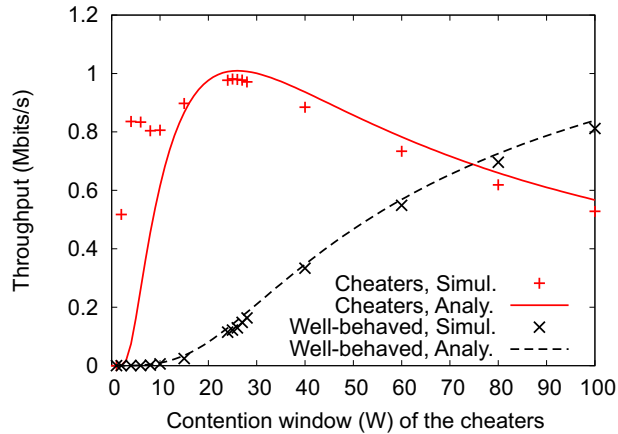


Figure 1.4: Throughput vs. contention window size of the cheaters (20 nodes, out of which 10 cheaters)

Since the problem  $\Pi_2$  is just a continuous version of the problem  $\Pi_1$ , i.e., we relax the integrality constraints on the variables  $W_i$ ,  $\forall i \in \mathcal{I}$ , we conjecture that an equivalent to Theorem 5 also holds for the problem  $\Pi_1$ . Observe that  $\tau_i = \tau^*$  ( $\forall i \in \mathcal{I}$ ) is not necessarily a feasible solution of the problem  $\Pi_1$ . However, if there exists an integer  $W$  such that  $\tau^* = 1/(W + 1)$ , then we know that vector  $(W_1 = W, \dots, W_I = W)$  is a unique solution to the problem  $\Pi_1$ . This follows from the fact that  $(W_1 = W, \dots, W_I = W)$  is a feasible solution to the problem  $\Pi_1$  and  $v_2 \geq v_1$ .

**Conjecture 1** *The problem  $\Pi_1$  admits a unique solution satisfying  $W_i = W^*$ , ( $W^* \in S_i$ ),  $\forall i \in \mathcal{I}$ .*

Observe that in this case the point  $(W_i = W^*)_{i \in \mathcal{I}}$  is Pareto optimal. This follows readily from the format of the problem  $\Pi_1$ . Thus, the solution  $(W_i = W^*)_{i \in \mathcal{I}}$  satisfies the symmetry ( $S$ ) and Pareto optimality ( $P$ ) Nash axioms (cf. Section 1.2.2). It is also easily seen that it satisfies the other Nash axioms, namely,  $IUO$ ,  $IUU$  and  $IAA$ ; this is also implied by the format of the problem  $\Pi_1$ .

In order to find the optimal value of the contention window  $W^*$ , on Figure 1.4 we plot the average aggregated throughput (the system throughput) obtained by 10 cheaters, all of which use the *same* contention window size; the simulation setup was described in Section 1.4.1 (Table 1.1). Note that in the simulations we take into account well-behaved nodes; they, however, do not affect that qualitative conclusions of the analytical treatment in this section. From this figure we can see that there exists a unique joint contention window size  $W^*$  maximizing the system throughput, which is consistent with the conclusion of Theorem 5 and Conjecture 1. A similar observation was already made by Bianchi in [19].

We conclude that the strategy profile  $(W_i = W^*)_{i \in \mathcal{I}}$  exhibits all the properties of a desirable point of operation in the CSMA/CA game  $G_{\text{CSMA/CA}}$ . In our context, this is significant since  $(W_i = W^*)_{i \in \mathcal{I}}$  is *not* a Nash equilibrium point (because,  $W_i^* > 1$ ,  $\forall i \in \mathcal{I}$ ) and as such might not be stable. Therefore, in the following section, we look at how to make the conjectured Pareto-optimal point  $(W_i = W^*)_{i \in \mathcal{I}}$  a Nash equilibrium point.



## 1.7 Multiple Stage CSMA/CA Game

Having determined the desirable point of operation  $(W_i = W^*)_{i \in \mathcal{I}}$ , we now intend to devise a strategy allowing the players to converge to this point. For this purpose, we make use of the theory of *repeated games* [40]. Repeated games capture the idea that a player can condition his future moves on the previous outcomes in the game. Using this model, we show how to make the point  $(W_i = W^*)_{i \in \mathcal{I}}$  a Nash equilibrium of the game  $G_{\text{CSMA/CA}}^\infty$ . We also devise a simple distributed algorithm that leads the players to this equilibrium point.

### 1.7.1 Nash Equilibria of the Repeated Game

Essentially, the multiple stage (or repeated) CSMA/CA game is defined as the game  $G_{\text{CSMA/CA}}$  played repeatedly  $T$  times. In our study, we consider an infinitely repeated game, that is,  $T \rightarrow \infty$ . We denote the repeated CSMA/CA game by  $G_{\text{CSMA/CA}}^\infty$ . In this new setting, the utility function of every player  $i \in \mathcal{I}$  changes to

$$u_i^\infty = \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T u_i^t(\tau_i^t, \tau_{-i}^t) \quad (1.42)$$

where  $u_i^t(\tau_i^t, \tau_{-i}^t)$  denotes a stage  $t$  payoff for the player  $i$ ; the lim inf in this expression is in response to the fact that some infinite sequences of stage payoffs do not have well-defined average values. One of the reasons that we do not use the *discounting criteria*, where “impatient” players discount future payoffs, is that, as we show in Section 1.8, the players in our game converge reasonably fast to a game equilibrium. Therefore, it is reasonable to assume that the players are “completely patient” (no discounting).

For mathematical convenience, we assume  $W_i^t$  (that is,  $\tau_i^t$ ) to be a continuous variable for every player  $i \in \mathcal{I}$ , and for all  $t = \{1, \dots, T\}$ . Moreover,  $\tau_i^t \in [0, 1]$ ,  $\forall i \in \mathcal{I}$ , ( $t = \{1, \dots, T\}$ ).

Let us define the following *penalty functions* for every player  $i \in \mathcal{I}$

$$p_i(\tau_i, \tau_{-i}) = \begin{cases} \varphi_i(\tau_i, \tau_{-i}), & \tau_i \in (\bar{\tau}, 1]; \\ 0, & \tau_i \in [0, \bar{\tau}], \end{cases} \quad (1.43)$$

where  $\bar{\tau} \in (0, 1)$  represents a *targeted* equilibrium point and  $\varphi_i(\tau_i, \tau_{-i})$  satisfies

$$\varphi_i(\tau_i, \tau_{-i}) > 0 \text{ and } \frac{\partial}{\partial \tau_i} \varphi_i(\tau_i, \tau_{-i}) > \frac{\partial}{\partial \tau_i} r_i^{(c)}(\tau_i, \tau_{-i}), \quad \forall \tau_i \in (\bar{\tau}, 1] \text{ and } \tau_j < 1 \quad (j \in \mathcal{I} \setminus \{i\}). \quad (1.44)$$

Let us further define the players' per stage payoffs as

$$u_i^t(\tau_i^t, \tau_{-i}^t) = r^{(c)t}(\tau_i^t, \tau_{-i}^t) - p_i^t(\tau_i^t, \tau_{-i}^t), \quad \forall i \in \mathcal{I}. \quad (1.45)$$

We note here that any *penalizing mechanism* used to impose the penalty  $p_i^t$  on some player  $i$ , should be designed so that it does not bring any performance degradation to the players  $k \in \mathcal{I} \setminus \{i\}$ . A “nice” property of the single-channel single-collision domain CSMA/CA networks is that at any time instant only one station can successfully transmit. Therefore, in these networks, we can single out any player for punishment. In game theory, this property is known as *full dimensionality* [40].

**Lemma 2** *Let  $\tau_j^t < 1$ ,  $\forall j \in \mathcal{I} \setminus \{i\}$ . Then, the stage payoff function  $u_i^t(\tau_i^t, \tau_{-i}^t)$  has a unique maximizer  $\tau_i^t = \bar{\tau} \in (0, 1)$  for every stage  $t = \{1, \dots, T\}$ .*

*Proof:* Since  $\tau_k^t < 1, \forall k \in \mathcal{I} \setminus \{i\}$ , we have from the equation (1.21)

$$\frac{\partial}{\partial \tau_i^t} r_i^{(c)t}(\tau_i^t, \tau_{-i}^t) > 0 \quad (1.46)$$

for  $\tau_i^t \in [0, 1]$ . Therefore, on the interval  $[0, \bar{\tau}]$ ,  $\tau_i^t = \bar{\tau}$  is the unique maximizer of the payoff  $u_i^t(\tau_i^t, \tau_{-i}^t)$ . For the remaining interval  $(\bar{\tau}, 1]$  we have

$$\frac{\partial}{\partial \tau_i^t} u_i^t(\tau_i^t, \tau_{-i}^t) = \frac{\partial}{\partial \tau_i^t} r_i^{(c)t}(\tau_i^t, \tau_{-i}^t) - \frac{\partial}{\partial \tau_i^t} \varphi_i^t(\tau_i^t, \tau_{-i}^t) \stackrel{(1)}{<} 0,$$

where the inequality (1) follows from the condition (1.44). Therefore, on the interval  $(\bar{\tau}, 1]$ ,  $u_i^t(\tau_i^t, \tau_{-i}^t)$  is a strictly decreasing function in  $\tau_i^t$ , which concludes the proof.  $\square$

Lemma 2 implies that the strategy profile  $(\tau_i^t = \bar{\tau})_{i \in \mathcal{I}}$  is the unique Nash equilibrium of the constituent game  $G_{\text{CSMA/CA}}$  played in stage  $t$ . In order to study the equilibria of the repeated game  $G_{\text{CSMA/CA}}^\infty$ , we first define (informally) the notion of a *subgame-perfect equilibrium* (or *Subgame Perfect Nash Equilibrium (SPNE)*). Informally, a strategy profile  $(\tau_i^t)_{i \in \mathcal{I}, t = \{1, \dots, T\}}$  is a subgame perfect Nash equilibrium if it induces a Nash equilibrium in every *subgame* of  $G_{\text{CSMA/CA}}^\infty$ , that is, if  $(\tau_i^t)_{i \in \mathcal{I}, t = \{1, \dots, T\}}$  is a Nash equilibrium of  $G_{\text{CSMA/CA}}^\infty$  then  $(\tau_i^t)_{i \in \mathcal{I}, t = \{k, \dots, T\}}$  is a Nash equilibrium of the subgame  $G_{\text{CSMA/CA}}^{k, \infty}$  played from stage  $k$  on, for every  $k \in \{1, \dots, \infty\}$ . Note that with this notation  $G_{\text{CSMA/CA}}^\infty = G_{\text{CSMA/CA}}^{1, \infty}$ .

**Theorem 6** *A strategy profile  $(\tau_i^t = \bar{\tau})_{i \in \mathcal{I}, t = \{1, \dots, T\}}$  is a subgame perfect Nash equilibrium (SPNE) of the game  $G_{\text{CSMA/CA}}^\infty$ .*

*Proof:* For every  $k \in \{1, \dots, T\}$  and every player  $i \in \mathcal{I}$  the following holds

$$\begin{aligned} u_i^{k, \infty} &= \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=k}^T u_i^t(\tau_i^t, \tau_{-i}^t) \\ &\leq \liminf_{T \rightarrow \infty} \frac{1}{T} \sum_{t=k}^T \max_{\tau_i^t \in [0, 1]} \{u_i^t(\tau_i^t, \tau_{-i}^t)\} \\ &\stackrel{(1)}{=} u_i((\tau_j = \bar{\tau})_{j \in \mathcal{I}}), \end{aligned}$$

where (1) follows from Lemma 2. Therefore, by definition,  $(\tau_i^t = \bar{\tau})_{i \in \mathcal{I}, t = \{1, \dots, T\}}$  is a SPNE of  $G_{\text{CSMA/CA}}^\infty$ .  $\square$

Observe that  $(\tau_i^t = \bar{\tau})_{i \in \mathcal{I}, t = \{1, \dots, T\}}$  is not a unique SPNE under the averaging criterion given by (1.42). The reason is that any finite number of deviations by some player  $i$  from the equilibrium strategy  $\tau_i^t = \bar{\tau}$  becomes irrelevant under the averaging criterion (1.42). Still, the prevalent strategy of the player  $i$  should be  $\tau_i^t = \bar{\tau}$ , since otherwise his overall payoff will be strictly smaller than  $u_i((\tau_j = \bar{\tau})_{j \in \mathcal{I}})$ ; in  $G_{\text{CSMA/CA}}^\infty$ , any deviation from  $\bar{\tau}$  necessarily result in the smaller per stage payoff. We will see in Section 1.7.3 the importance of this ‘‘insensitivity to finite losses’’.

The following corollary is a simple implication of the penalty functions  $p_i$ , ( $i \in \mathcal{I}$ ), defined by (1.43). This result is reminiscent of the *Nash Folk theorem* [40].

**Corollary 1** *Any strategy profile  $(\tau_i^t = \tau)_{i \in \mathcal{I}, t = \{1, \dots, T\}}$ , such that  $\tau \in (0, 1)$ , can be made a SPNE.*

In our context, this result is important as we want to make the Pareto optimal point  $(W_i = W^*)_{i \in \mathcal{I}}$ , i.e., the corresponding channel access probability profile  $(\tau_i = 2/(1 + W^*))_{i \in \mathcal{I}}$ , a Nash equilibrium.

### 1.7.2 Practical Penalty Function

Let us consider two arbitrary players  $k$  and  $i$  from set  $\mathcal{I}$ . Let us assume that player  $k$  calculates the penalty  $p_i$  to be inflicted on player  $i$  as follows

$$p_i(\tau_i, \tau_{-i}) = \begin{cases} r_i^{(c)}(\tau_i, \tau_{-i}) - r_k^{(c)}(\tau_i, \tau_{-i}), & \text{if } r_i^{(c)}(\tau_i, \tau_{-i}) > r_k^{(c)}(\tau_i, \tau_{-i}); \\ 0, & \text{otherwise.} \end{cases} \quad (1.47)$$

It is easily seen that the penalty function (1.47) has essentially the same format as the penalty function given by (1.43) and (1.44). To see this, using the notation of the definition in (1.43), we define  $\varphi_i(\tau_i, \tau_{-i}) \stackrel{\text{def}}{=} r_i^{(c)}(\tau_i, \tau_{-i}) - r_k^{(c)}(\tau_i, \tau_{-i})$ , where  $r_i^{(c)}(\tau_i, \tau_{-i}) > r_k^{(c)}(\tau_i, \tau_{-i})$ . Observe that the condition  $r_i^{(c)}(\tau_i, \tau_{-i}) > r_k^{(c)}(\tau_i, \tau_{-i})$  in (1.47) is equivalent to  $\tau_i > \tau_k$  when  $\tau_j < 1$ ,  $\forall j \in \mathcal{I} \setminus \{i\}$ . Finally, for  $\tau_j < 1$ ,  $\forall j \in \mathcal{I} \setminus \{i\}$ , we have

$$\frac{\partial}{\partial \tau_i} \varphi_i(\tau_i, \tau_{-i}) = \frac{\partial}{\partial \tau_i} r_i^{(c)}(\tau_i, \tau_{-i}) + \left| \frac{\partial}{\partial \tau_i} r_k^{(c)}(\tau_i, \tau_{-i}) \right| \stackrel{(1)}{>} \frac{\partial}{\partial \tau_i} r_i^{(c)}(\tau_i, \tau_{-i}),$$

where (1) follows from the fact that  $\tau_j < 1$ ,  $\forall j \in \mathcal{I} \setminus \{i\}$ .

Therefore, we can apply Lemma 2 to conclude that the unique maximizer of the player  $i$ 's (single stage) payoff  $u_i(\tau_i, \tau_{-i})$  is  $\tau_i = \tau_k$ . In the context of the two players  $i$  and  $k$ , a very important property of the penalty function is that it results in the same throughputs for both player  $i$  and player  $k$ ; i.e.,  $\tau_i = \tau_k$  implies that players  $i$  and  $k$  will receive the same throughputs.

Inspired by the penalty functions (1.47), we have designed a simple penalizing scheme, in which the packets of the *noncooperative* player are *selectively jammed* for a short duration of time,  $T^{jam}$ , by the other players in the system. By the “noncooperative player” we mean the player that deviates from the given equilibrium point. Suppose that a player  $k \in \mathcal{I}$  detects the presence of a noncooperative player  $i \in \mathcal{I}$ . Thereafter, if the player  $k$  listens to a transmitted packet corresponding to the player  $k$ , it switches to transmission mode and *jams* enough bits so that the packet cannot be properly recovered at the receiver.

Let the throughput obtained by the two considered players over the last *observation window*,  $T^{obs}$ , be  $r_i^{(c)}$  and  $r_k^{(c)}$ , respectively, where  $r_i^{(c)} > r_k^{(c)}$ . As we saw above, the penalty function (1.47) aims at making the throughputs received by the players  $i$  and  $k$  equal. We denote with  $r_x^{(c)}(t)$  the instantaneous throughput of the given player  $x$ . The *average throughput* received by the players  $i$  and  $k$  should be the same over the total time duration of  $T^{obs} + T^{jam}$ , that is,

$$\begin{aligned} \frac{1}{T^{obs} + T^{jam}} \int_t^{t+T^{obs}+T^{jam}} r_k^{(c)}(t) dt &= \frac{1}{T^{obs} + T^{jam}} \int_t^{t+T^{obs}+T^{jam}} r_i^{(c)}(t) dt \\ &\stackrel{(1)}{=} \frac{1}{T^{obs} + T^{jam}} \int_t^{t+T^{obs}} r_i^{(c)}(t) dt, \end{aligned} \quad (1.48)$$

where (1) follows from the fact that the player  $k$  jams the player  $i$  during the period  $T^{jam}$ . Let us denote the average throughput over a time period  $P$  starting at time instant  $t$  by  $\bar{r}(t, P)$ , that is,

$$\bar{r}(t, P) \stackrel{\text{def}}{=} \frac{1}{P} \int_t^{t+P} r(t) dt.$$

Then, from the expression (1.48) we obtain

$$T^{jam} = T^{obs} \frac{\bar{r}_i^{(c)}(t, T^{obs}) - \bar{r}_k^{(c)}(t, T^{obs})}{\bar{r}_k^{(c)}(t + T^{obs}, T^{jam})}. \quad (1.49)$$

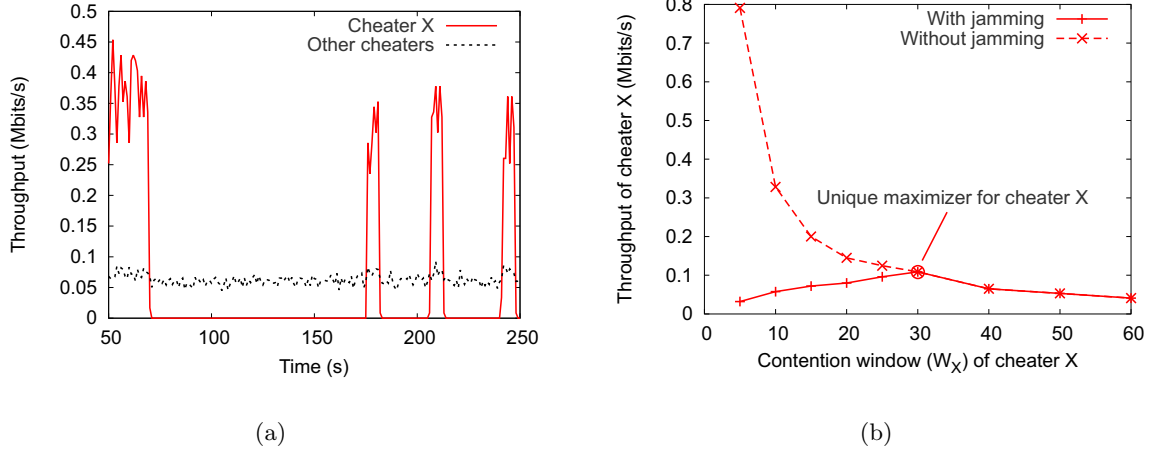


Figure 1.5: Realization of the penalty function  $p_i(\tau_i, \tau_{-i})$  by the selective jamming: (a) Throughputs (payoffs) obtained by the cheaters over time in the presence of the noncooperative cheater  $X$  and selective jamming mechanism; (b) Unilateral deviation by the cheater  $X$  with and without the penalty mechanism.

We note that  $T^{jam} < \infty$ , except in the case when  $\bar{r}_k^{(c)}(t + T^{obs}, T^{jam}) = 0$ . But the case  $\bar{r}_k^{(c)}(t + T^{obs}, T^{jam}) = 0$  never happens under the penalty functions (1.47) if all the players are rational (and the number of players is finite); by the Theorem 6 there are strictly better outcomes than zero for every player. It is also interesting to observe that the noncooperative player  $i$  minimizes  $T^{jam}$  by playing  $\tau_i = 0$  during the period  $T^{jam}$ . This is because  $\partial r_k^{(c)} / \partial \tau_i < 0$ , when  $\tau_j < 1$ ,  $\forall j \in \mathcal{I} \setminus \{i\}$ , and therefore  $\bar{r}_k^{(c)}(t + T^{obs}, T^{jam})$  gets larger.

We have implemented the jamming mechanism in *ns-2*. The simulation setup is the same as in Section 1.4.1 with  $N = 20$  and  $I = 10$ . We randomly pick up a cheating player, designated as cheater  $X$ , and fix his contention window size to be 10. The contention window size for all the other cheaters in the system is fixed to the point  $\bar{W} = 30$ , (i.e., the corresponding  $\bar{\tau}$ ). We use an observation window size,  $T^{obs}$ , of 20 seconds. Cheater  $X$  gets *detected* by the other cheaters in the network and is penalized for his deviation. We describe the detection mechanism in Section 1.8.1. On Figure 1.5(a) we plot the throughput obtained by the cheaters in the system over time, with and without the penalizing scheme. As can be observed from Figure 1.5(a), cheater  $X$  is detected and is penalized for his deviation. When penalized, the cheater  $X$ 's throughput drops to zero. Observe from this figure the dependency of the period  $T^{jam}$  on the observation period  $T^{obs}$ ; for better system efficiency,  $T^{obs}$  should be kept short (much shorter than 20 seconds as used in our simulations).

Figure 1.5(b) plots the average throughput obtained by cheater  $X$ , when it unilaterally deviates from the given equilibrium point  $\bar{W} = 30$ . The results are averaged over a duration of 1000 seconds. As can be observed from Figure 1.5(b), after the introduction of the detection and penalizing mechanism, cheater  $X$  achieves maximum throughput by operating at the given equilibrium point  $\bar{W}$ , i.e.,  $\bar{\tau}$ , which is consistent with the result of Lemma 2. Thus, any unilateral deviation from this point brings less payoff to the cheater  $X$ . Therefore, by definition,  $\bar{W}$  is a unique Nash equilibrium of the single stage game.

### 1.7.3 Equilibrium Coordination Algorithms

We first describe a simple algorithm that leads the players to a unique equilibrium point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$  (not necessarily Nash) of the given stage. Then, we show how to use this algorithm to make the Pareto optimal point  $(\tau_i = 2/(1 + W^*))_{i \in \mathcal{I}}$  a subgame perfect Nash equilibrium.

The main idea is that one player acts as a *coordinator* by inflicting penalties on other players that receive higher payoffs (throughputs). We assume that initially  $\tau_i < 1, \forall i \in \mathcal{I}$ . Let us denote the coordinator player with  $k$ . The initial access probability of the coordinator is  $\tau_k = \bar{\tau} \leq 1 - \varepsilon$ , where  $0 < \varepsilon \ll 1$  (arbitrarily small). Now, consider the following adaptation algorithm used by every player  $i \in \mathcal{I} \setminus \{k\}$ :

$$\frac{d\tau_i}{dt} = \frac{\partial u_i(\tau_i, \tau_{-i})}{\partial \tau_i} \quad \text{while ensuring} \quad \tau_i \leq 1 - \varepsilon. \quad (1.50)$$

This adaptation algorithm is inspired by Lemma 2, from which we know that the payoff functions  $u_i(\tau_i, \tau_{-i}) = r_i^{(c)}(\tau_i, \tau_{-i}) - p_i(\tau_i, \tau_{-i}), \forall i \in \mathcal{I}$  admit the unique maximizer  $\tau_i = \bar{\tau}$ . Note that the algorithm (1.50) implies that each player  $i \in \mathcal{I} \setminus \{k\}$  simply adjusts his access probability so that his payoff is maximized.

**Theorem 7** *For any initial channel access probability point  $(\tau_i)_{i \in \mathcal{I}}$ , such that  $\tau_i \leq 1 - \varepsilon, \forall i \in \mathcal{I}$ , the algorithm (1.50) converges to  $\tau_i = \bar{\tau}, \forall i \in \mathcal{I}$ .*

*Proof:* Let us denote with  $\tau$  the point  $\tau = (\tau_i - \bar{\tau})_{i \in \mathcal{I} \setminus \{k\}}$ . Following Lyapunov stability theory [94], we first define a function

$$V(\tau) = \frac{1}{2} \sum_{i \in \mathcal{I} \setminus \{k\}} (\tau_i - \bar{\tau})^2.$$

Note that  $V(\tau)$  is a positive definite function, since  $V(\tau) > 0$  except for  $(\tau_i = \bar{\tau})_{i \in \mathcal{I} \setminus \{k\}}$  (i.e.,  $V(\mathbf{0}) = 0$ ). Next we take the time derivative of  $V(\tau)$  to obtain

$$\begin{aligned} \frac{dV(\tau)}{dt} &= \sum_{i \in \mathcal{I} \setminus \{k\}} (\tau_i - \bar{\tau}) \frac{d\tau_i}{dt} \\ &\stackrel{(1)}{=} \sum_{i \in \mathcal{I} \setminus \{k\}} (\tau_i - \bar{\tau}) \frac{\partial u_i(\tau_i, \tau_{-i})}{\partial \tau_i} \\ &\stackrel{(2)}{=} \sum_{i \in \mathcal{I} \setminus \{k\}} (\tau_i - \bar{\tau}) \times \begin{cases} \frac{\partial r_k^{(c)}(\tau_i, \tau_{-i})}{\partial \tau_i}, & \text{if } \tau_i > \bar{\tau}; \\ \frac{\partial r_i^{(c)}(\tau_i, \tau_{-i})}{\partial \tau_i}, & \text{if } \tau_i \leq \bar{\tau} \end{cases} \\ &\begin{cases} \stackrel{(3)}{<} 0 & \text{if } \tau_i \neq \bar{\tau}; \\ \stackrel{(4)}{=} 0, & \text{if } \tau_i = \bar{\tau}, \end{cases} \end{aligned}$$

where (1) follows from the definition (1.50), (2) follows from the definition of the penalty functions (1.47), (3) follows from  $\partial r_k^{(c)}(\tau_i, \tau_{-i})/\partial \tau_i < 0$  and  $\partial r_i^{(c)}(\tau_i, \tau_{-i})/\partial \tau_i > 0$  (which holds since  $\tau_i < 1, \forall i \in \mathcal{I}$ ), and (4) follows from the fact that  $\partial r_i^{(c)}(\tau_i, \tau_{-i})/\partial \tau_i < \infty$ . Therefore,  $V(\tau)$  is a Lyapunov function for the system of  $I - 1$  differential equation(1.50), that is, this system converges to the point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I} \setminus \{i\}}$  that is a globally asymptotically stable equilibrium point [94].  $\square$

Observe, however, that  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$  is not a Nash equilibrium point, since the coordinator has an incentive to deviate from this point; the coordinator is not penalized. For example, by misusing

the penalization mechanism in such a way that the coordinator forces the other players to play the strategy  $\bar{\tau}' \ll \bar{\tau}$ , the coordinator can increase his own payoff  $u_k(\tau_k, \tau_{-k})$ . A potential remedy to this problem is to let the other players to act as coordinators after some finite time (sufficiently long for the above algorithm to converge), if they observe that the system did not converge to some common point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$ . Recall that the players are not sensitive to finite-time losses under the averaging criterion (1.42). In this case, if the original coordinator is not adaptive, the whole system will be pulled to the equilibrium point  $\tau^{(1)} = (\tau_k = \bar{\tau}, (\tau_i = \bar{\tau}')_{i \in \mathcal{I} \setminus \{k\}})$ , where  $\bar{\tau}' \in \{\arg \min_{i \in \mathcal{I} \setminus \{k\}} \{\tau_i\}\}$ . If the player  $k$  is adaptive, the system will converge to the equilibrium point  $\tau^{(2)} = (\tau_i = \bar{\tau}')_{i \in \mathcal{I}}$  (Lemma 2). Therefore,  $u_k(\tau^{(2)}) > u_k(\tau^{(1)})$ . In this way, the original coordinator will have an incentive to misbehave against the other players, only if the point  $\tau^{(2)}$  gets him larger payoff than  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$ , that is, if

$$u_k(\tau^{(2)}) > u_k((\tau_i = \bar{\tau})_{i \in \mathcal{I}}) .$$

But in this case, the payoff of every other player increases as well, since  $(\tau_i = \bar{\tau}')_{i \in \mathcal{I} \setminus \{k\}}$  is an equilibrium point according to Theorem 7. Therefore, by making his decisions in a pure selfish way, the player  $k$  actually acts in the interest of the overall system.

To motivate further the coordinator to follow the strategy prescribed by the coordination algorithm (1.50), we next propose a simple adaptive algorithm that is run by the coordinator upon the algorithm (1.50) has converged to some point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$  in the given stage, and which increases the coordinator's payoff in each stage until all the players settle down on the Pareto optimal equilibrium point  $(\tau_i^* = 2/(1 + W^*))_{i \in \mathcal{I}}$ . This is the unique point that maximizes the payoffs of each player simultaneously (see Figure 1.4), i.e., for every player  $i \in \mathcal{I}$  we have

$$u_i \left( \left( \tau_i = \frac{2}{1 + W^*} \right)_{i \in \mathcal{I}} \right) \geq u_i((\tau_i = \tau)_{i \in \mathcal{I}}) \quad \text{for any } \tau \in [0, 1] .$$

Therefore, the coordinator has no incentive to deviate from this point. Since, in addition, the coordinator inflicts penalties on all the other players, the Pareto optimal point  $(\tau_i^* = 2/(1 + W^*))_{i \in \mathcal{I}}$  is a subgame perfect Nash equilibrium (SPNE) according to Theorem 6. Note that the threat that the other players will become coordinators themselves if the system does not stabilize, in a prescribed finite time, on some equilibrium  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$ , is crucial for the above result about the SPNE to hold.

After the players stabilize on some stage equilibrium point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I}}$ , the coordinator  $k$  changes his access probability  $\tau_k$  according to the following gradient based algorithm

$$\frac{d\tau_k}{dt} = \frac{\partial u_k((\tau_i = \tau_k)_{i \in \mathcal{I}})}{\partial \tau_k} = \frac{\partial r_k^{(c)}((\tau_i = \tau_k)_{i \in \mathcal{I}})}{\partial \tau_k} . \quad (1.51)$$

The algorithm (1.51) simply tries to pinpoint the unique maximizer  $\tau^* = 2/(1 + W^*)$  of the function  $r_k((\tau_i = \tau_k)_{i \in \mathcal{I}})$ , that is, of the aggregate throughput function  $r^{agg} = I r_k((\tau_i = \tau_k)_{i \in \mathcal{I}})$  (see Figure 1.4). That this algorithm converges follows easily from the fact that the positive definite function

$$V(\tau) = \frac{1}{2} (\tau_k - \tau^*)^2 ,$$

with  $\tau = \tau_k - \tau^*$ , is a Lyapunov function for the differential equations in (1.51). Indeed,

$$\begin{aligned} \frac{dV(\tau)}{dt} &= (\tau_k - \tau^*) \frac{d\tau_k}{dt} \\ &= (\tau_k - \tau^*) \frac{\partial r_k^{(c)}((\tau_i = \tau_k)_{i \in \mathcal{I}})}{\partial \tau_k} \\ &\begin{cases} < 0 & \text{if } \tau_k \neq \tau^*; \\ = 0, & \text{if } \tau_k = \tau^*. \end{cases} \end{aligned}$$

Therefore, the overall coordination procedure can be seen as switching between the two algorithms (1.50) and (1.51). The coordinator chooses some initial value  $\tau_k = \bar{\tau}$  and begins to penalize the other players with  $r_i^{(c)} > r_k^{(c)}$ ,  $i \neq k$ . The other players act in self-interest and run the algorithm (1.50) until they all stabilize at the stage equilibrium point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I} \setminus \{k\}}$ . After spending some finite time on this point (to let the other players learn that the system has reached the point  $(\tau_i = \bar{\tau})_{i \in \mathcal{I} \setminus \{k\}}$  and thus to avoid being penalized by them), the coordinator updates his strategy according to the algorithm (1.51). In turn, the other players start running the algorithm (1.50) again. This procedure eventually converges to the Pareto optimal SPNE point  $(\tau_i = 2/(1 + W^*))_{i \in \mathcal{I}}$ .

## 1.8 Implementation

In this section, we will build a comprehensive, distributed and efficient equilibrium coordination protocol based on the theoretical insights from Section 1.7. We saw that the key building block for the model of repeated games is the penalization mechanism. We have already elaborated a practical penalization mechanism in Section 1.7.2. The penalization mechanism, however, relies on the ability of the players to estimate the difference in their payoffs. In order to empower the players with this ability, we first develop an appropriate *detection mechanism*. Then, we describe how the players should react once they are penalized. We call the scheme followed by the penalized nodes an *adaptive strategy*. Finally, we put together all the basic building blocks and simulate the behavior of such a comprehensive coordination algorithm.

### 1.8.1 Detection Mechanism

In our approach, each cheating node (player) measures the throughput of each other node<sup>2</sup>, including itself. This is indeed feasible due to the broadcast nature of the wireless medium. If a cheater observes a difference in throughput with some other node, it characterizes that node as a deviating cheater. Let  $r_i$  and  $r_j$  be the measured throughput of nodes  $i$  and  $j$ , respectively. Due to the inherent short-time unfairness of the IEEE 802.11 MAC protocol [57], and in order to increase the efficiency of the detection mechanism, we use two parameters: the observation time-window size  $T^{obs}$  and the tolerance margin  $\epsilon$ , in percentage of throughput. After measuring the throughput of each node for  $T^{obs}$  seconds, cheater  $i$  concludes that cheater  $j$  is *deviating* whenever the throughput of node  $j$  exceeds the throughput of node  $i$ , that is, whenever

$$\frac{r_j}{r_i} > 1 + \epsilon .$$

---

<sup>2</sup>We deliberately use the word *node* (and not cheater), since in reality well-behaved nodes may be present as well (even though we neglect them in the analysis).

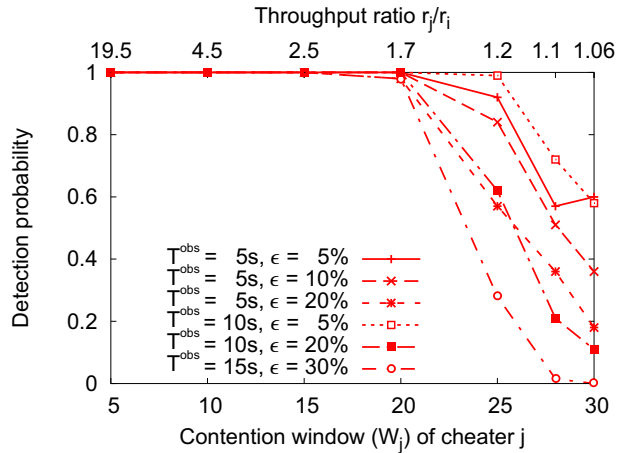


Figure 1.6: Performance of cheating detection based on throughput measurements

We have implemented this detection mechanism in *ns-2*, with  $N = I = 30$  nodes. We vary the contention window size ( $W_j$ ) of a single node  $j$ , and set others' contention window sizes to 30 (ie.,  $W_k = 30, \forall k \in \mathcal{I} \setminus \{j\}$ ).

Figure 1.6 shows the performance of the detection mechanism for different values of  $T^{obs}$  and  $\epsilon$ . The probability of false positives corresponds to the detection probability with  $W_j = 30$ ; at this point, cheater  $j$  uses a contention window value equal to that of node  $i$ , but still gets a higher throughput,  $r_j/r_i = 1.06$ , due to the IEEE 802.11 unfairness. Therefore, node  $j$  gets detected as deviating with positive detection probability. To reduce the false positives (at contention window size 30), one can consider large  $\epsilon$  values ( $> 10\%$ ). However, this comes at the expense of lower detection probabilities if cheater  $j$  uses contention window sizes slightly lower than 30. Similarly, large  $T^{obs}$  values ( $\geq 15s$ ) will reduce the effect of the inherent IEEE 802.11 unfairness, and therefore the corresponding false positives. This also comes at the expense of lower detection probabilities if cheater  $j$  uses contention window sizes slightly lower than 30. Therefore, choosing appropriate values for  $T^{obs}$  and  $\epsilon$  is crucial for both our detection mechanism and the overall system performance. For very low contention window sizes of cheater  $j$  ( $W_j \leq 20$ ), the throughput ratio  $r_j/r_i$  is much larger than  $1 + \epsilon$ , making the detection of the cheater  $j$ 's deviation easy.

A recently proposed detection mechanism [90], based on calculating the average backoff used by the nodes, can be used in the case of heterogeneous conditions among the cheaters in the system. Although the approach in [90] is more appropriate for misbehaving detection at the MAC layer, we consider here the throughput-based detection for simplicity of implementation. Our equilibrium coordination algorithm can be easily adapted to any detection mechanism used.

## 1.8.2 Adaptive Strategy

Inspired by the adaptive strategy of algorithm (1.50), we have implemented the following adaptive strategy. When cheater  $i$  observes that he is being jammed (penalized) during some period  $\Delta$ , he gradually increases his contention window by steps of size  $\gamma$ . Note that a cheater can easily decide whether he is being jammed by observing his own throughput. The choice of  $\Delta$  determines the efficiency of the system. A high value of  $\Delta$  might let a non-cooperative cheater escape from being penalized. However, choosing a small value of  $\Delta$  might magnify the effect of a possible misdetection



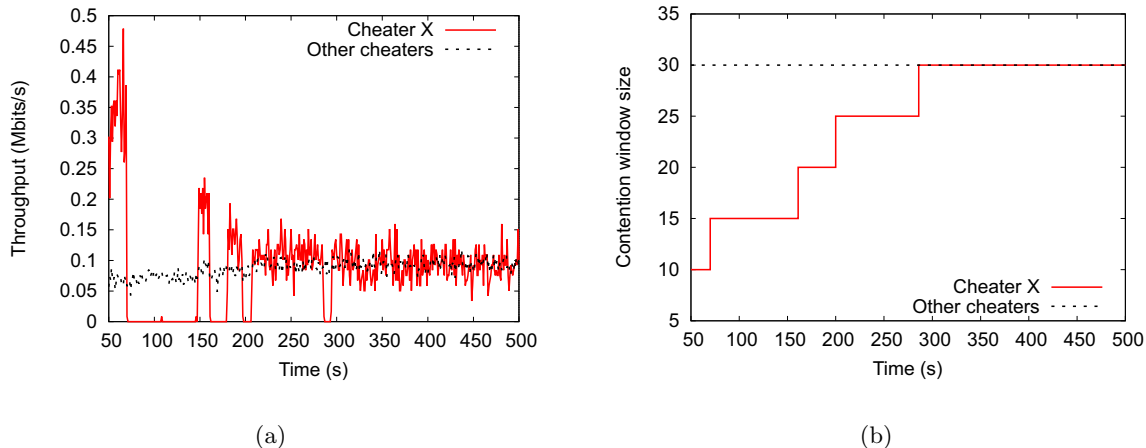


Figure 1.7: Performance of the system with the adaptive strategy: (a) Throughput of the cheaters over time; (b) Contention window size of the cheaters over time.

Table 1.2: Throughput obtained by different nodes (bytes/s)

	Strategy	
	Non-adaptive	Adaptive
Cheater $X$	7650	11577
Other cheaters	7826	11448
Well-behaved nodes	1286	2318

by unnecessarily causing a cheater to increase his contention window size. This will eventually lead the whole system towards an inefficient point of operation. The choice of the step size,  $\gamma$ , offers a tradeoff between convergence time and efficiency: If we increase the contention window in large steps, although the system will stabilize in less time, the point of operation might be far away from the Pareto-optimal point ( $W^*$ ), resulting in an inefficient system and vice-versa.

We have implemented this adaptive strategy in *ns-2*. The simulation setup is the same as in the previous section ( $N = 20, I = 10, W^* = 30$ ). We randomly pick up a cheater, designated as node  $X$ , and fix his initial contention window size to 10. The contention window size for all the other cheaters in the system is fixed to  $W^*$ . We fix  $\Delta$  to be 5 seconds and  $\gamma$  to be 5. Figure 1.7(a) plots the obtained throughput by different cheaters in the system over time. Figure 1.7(b) plots the evolution of contention window size of node  $X$  over time. One can observe how node  $X$  adapts its contention window size by following the adaptive strategy and eventually converging to a window size of 30, equal to  $W^*$ . Thus the other cheaters in the system are successful in guiding the deviating cheater to the desired equilibrium point.

Table 1.2 summarizes the throughput averages obtained by different nodes over a time interval of 1000 seconds. As can be observed from Table 1.2, the jamming and detection mechanism combined with the adaptive strategy, besides being fair to all the cheaters in the system, is also the most

Table 1.3: Throughput obtained by different nodes (bytes/s) with multiple levels of misbehavior

	Strategy	
	Non-adaptive	Adaptive
Cheater $X$	2843	10356
Cheater $Y$	2686	10185
Cheater $Z$	2565	10239
Other cheaters	2544	10172
Well-behaved nodes	270	1981

efficient.

Finally, we evaluate the performance of our protocol (in *ns-2*) for a scenario consisting of multiple levels of misbehavior in the system. The simulation setup is the same as above ( $N = 20, I = 10, W^* = 30$ ). We randomly pick up three cheaters, designated as node  $X, Y$  and  $Z$  respectively. We fix their contention window sizes to be 5, 10 and 15, respectively. The contention window size for all the other cheaters in the system is fixed to  $W^*$ . Table 1.3 summarizes the average throughput obtained by different nodes over an interval of 1000 seconds. As can be observed from Table 1.3, the jamming mechanism combined with the adaptive strategy results in an optimal and fair performance, even with multiple levels of misbehavior in the system. As we predicted in Section 1.7, the deviating cheaters (players)  $X, Y$  and  $Z$  clearly have an incentive to adapt upon being penalized. In the same way, the other cheaters have an incentive to penalize the other cheaters.

### 1.8.3 Reaching the Pareto-optimal Point

An accurate implementation of detection, penalizing and adaptive strategy will lead the nodes to reach a stage equilibrium point,  $(W_i = \overline{W})_{i \in \mathcal{I}}$ . However, the intention is to reach the Pareto optimal point  $(W_i = W^*)_{i \in \mathcal{I}}$ . As we described in Section 1.7.3, this can be achieved by alternating between the algorithms (1.50) and (1.51). Inspired by this approach, we have implemented the following distributed coordination algorithm.

At the onset of the system,  $(W_i = W^{init})_{i \in \mathcal{I}}$  for all cheaters. Every cheater sets up a random timer (in our simulations this corresponds to a random value between 0 and 20 seconds) to increase his contention window by step size,  $\gamma$ . One of the cheaters, say  $X$ , will eventually increase his contention window size to  $W_X^{init} + \gamma$ ; using the nomenclature of Section 1.7.3, cheater  $X$  plays the role of the coordinator. Based on the detection mechanism (Section 1.8.1), node  $X$  will conclude that all other cheaters in the system are deviating and will begin penalizing them. If a cheater observes that he is being penalized, he will disable the timer, and use the adaptive strategy described in Section 1.8.2. Eventually the system will stabilize, when  $W_i = W_i^{init} + \gamma$  for all cheaters. This is guaranteed by Theorem 7.

The cheaters realize that they have reached a new stable point of operation, when they all begin enjoying the same throughput (in our implementation, the cheaters remain at this stable point for 20 seconds before continuing the search for  $W^*$ ). At this point in time, every cheater  $i \in \mathcal{I}$

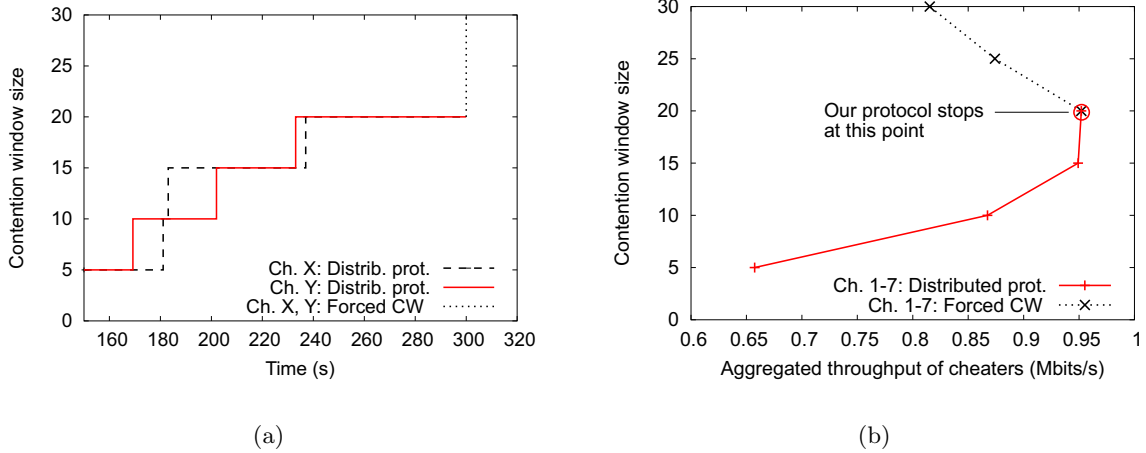


Figure 1.8: Performance of the distributed coordination protocol, with  $N = 20$  and  $I = 7$  (the axes in (b) are swapped for the convenience of matching them with (a)): (a) Evolution of the contention windows; (b) Contention window vs. Average throughput.

compares his throughput at  $W_i = W_i^{init} + \gamma$  with the throughput at  $W_i = W_i^{init}$ ; if he observes a decrease in his throughput, he will terminate the search for  $W^*$ . Otherwise he again sets up the random timer to increase his contention window size by  $\gamma$ . Note that this step is reminiscent of the algorithm (1.51). Therefore, the proposed distributed protocol simply “climbs” up the left side of the aggregate throughput curve shown on Figure 1.4, until it hits the optimal value  $W^*$ .

We have implemented this protocol in *ns-2*. The simulation setup consists of 20 nodes and 7 cheaters ( $N = 20, I = 7$ ). The cheaters initialize their contention window sizes to 5 ( $(W_i^{init} = 5)_{i \in \mathcal{I}}$ ). The cheaters continue their search for  $W^*$  only if they see an increase of 10% or more in their throughput from the last stable point of operation. Figure 1.8(a) plots the sample evolution of the contention window for 2 cheaters,  $X$  and  $Y$ , in the system. Note that all of the cheaters follow a similar pattern and eventually converge to a window size of 20. We are unable to show their evolution in the same plot as it simply generates overlapping lines. Note also that the convergence time is relatively short, around 80 seconds for 7 cheaters (from  $t_{start} \approx 160$  to  $t_{end} \approx 240$ ; in these simulations we used the warm-up period of around 160 seconds).

Figure 1.8(b) plots the average throughput obtained by the cheaters at different contention window sizes. As can be seen from Figure 1.8(b), the throughput is maximized at  $(W_i = 20)_{i \in \mathcal{I}}$ . In reality, the cheaters will stabilize at  $(W_i = 20)_{i \in \mathcal{I}}$ . For completeness, we obtain the “dotted” curves in Figure 1.8 by deliberately forcing the cheaters to go beyond  $(W_i = 20)_{i \in \mathcal{I}}$ .

We next evaluate the performance of our protocol by varying the number of cheaters in the system. We run our protocol and measure the window size at which all the cheaters eventually converge. Thus, according to our protocol, this point of convergence is the Pareto-optimal point of operation. We evaluate the actual Pareto-optimal point ( $W^*$ ), under the same network settings, through *ns-2* simulations. We also evaluate the Pareto-optimal point ( $W^*$ ) analytically, using the Bianchi’s model. Figure 1.9 plots the obtained results. The results are averaged over 5 simulation runs. The results obtained by our distributed protocol closely match the analytical results obtained using Bianchi’s model. Note that the minimum resolution of our protocol is equal to the step size,  $\gamma = 5$ . As can be seen from Figure 1.9, the discrepancy is bounded by  $\pm\gamma$ , which clearly proves the

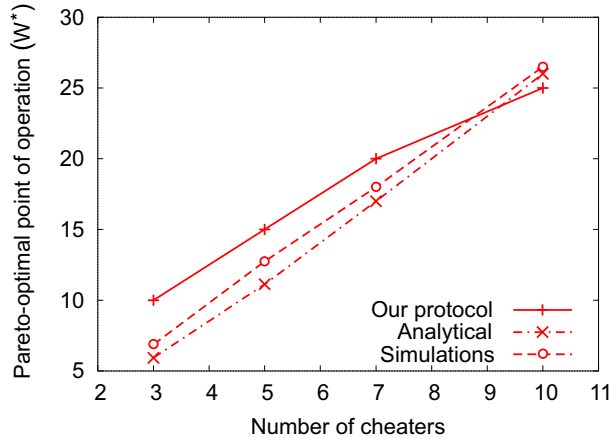


Figure 1.9: Variation of the Pareto optimal point  $W^*$  with the number of cheaters.

efficiency of our distributed protocol.

The protocol operates in a completely distributed manner, without requiring any *a priori* knowledge about the optimal point of operation or of the total number of nodes/cheaters in the system. However, we rely on the fact that the numbers of nodes and of cheaters does not change in the system. In more dynamic networks, where new nodes/cheaters can enter or existing nodes/cheaters can leave the system, we propose that cheaters time out periodically and re-run the whole protocol from the beginning.

## 1.9 Related Work

The problem of non-cooperative nodes in wireless (wired) networks has been widely addressed on the network layer, whereas little work has been done on the MAC layer. MacKenzie and Wicker [76] study the problem of selfish users in Aloha from a game-theoretic point of view. They analyze the stability of the system (Nash equilibrium), and calculate the transmission probabilities that optimize each node's throughput. They assume however that all nodes have the same transmission rates and costs. Moreover, every node has an *a priori* knowledge about the total number of nodes in the system. Altman et al. [12] reconsider the same Aloha "game" with partial information, where the transmission probability is adapted according to collision feedback only. They consider two frameworks: team work and non-cooperative game. Jin and Kesidis [53] study non-cooperative equilibria of Aloha networks for heterogeneous users.

For IEEE 802.11, Kyasanur and Vaidya [64] propose that the receiver assigns the backoff value to be used by the sender, so the former can detect any misbehavior of the latter. If the sender deviates from the assigned value, it will be assigned high backoff values on the next round to compensate its deviation. As mentioned by the authors, this mechanism has several limitations such as the possible collusion between sender and receiver, and the fundamental change to the protocol. Konorski [58] proposes a misbehaviour-resilient backoff algorithm that exhibits the same drawback: it requires to change the current protocol.

The Nash bargaining framework has already been proposed for fair bandwidth allocation for elastic services in wired networks by Yaïche et. al. [113]. The important difference between the Nash bargaining framework (the framework used in [113]) and the CSMA/CA game is that the set of feasible payoffs  $R$  is neither compact nor convex in the CSMA/CA game.

Game theory has been applied in the study of optimal routing [83, 59, 65], congestion control [113], power control [92, 10], as well as incentive engineering in wireless access networks [74].

## 1.10 Summary

In this chapter we have addressed the problem of cheating in single collision domain CSMA/CA networks. For this purpose, we have developed a game-theoretical model and verified our findings by appropriate simulations. We have made several contributions. First, we have provided a formalism for the systematic study of rational cheating in CSMA/CA networks. Second, we have studied the simple cases (i) of a single cheater and (ii) of several cheaters acting without restraint. Third, we have shown that the Nash Bargaining Framework (and the Nash Bargaining Solution) is applicable and a useful tool to address resource allocation problems on the MAC layer of wireless networks, even in the face of non-convexity and non-compactness of feasible payoff sets. Using the Nash bargaining framework, we have identified the Pareto optimal point of operation of a network with multiple cheaters. Fourth, using the theory of repeated (multistage) games, we have shown how it is possible to transform this Pareto optimal point into a Subgame Perfect Nash Equilibrium. Fifth, we have shown that smart cheaters can collectively find this point. We believe these contributions to be very relevant in wireless networks.

In terms of future work, we envision extending the game theoretic analysis used in this chapter to wireless networks of general topology.



## Chapter 2

# Wormhole Defense: New Anti-Jamming Techniques in Sensor Networks

### 2.1 Introduction

In this chapter, we investigate event-masking attacks on sensor networks, whereby an adversary prevents events detected by (a subset of) sensors from being reported to the sink (network operator). We study scenarios in which the attacker masks events by stealthily jamming an appropriate subset of the network nodes. Timely detection of such stealth attacks is particularly important in scenarios in which sensors use reactive schemes to communicate events to the network sink [111].

Event-masking attacks result in a *coverage paradox*: in spite of the fact that an event is sensed by one or several nodes (and the sensor network is fully connected), the network operator cannot be informed about the event on time (see Figure 2.1). We will explain that the solution to this problem is far from trivial: proactive schemes, in which sensors spend their time (and battery) assessing the state of their communication links are clearly suboptimal; likewise, reporting all observed events is inappropriate, as it would generate many false alarms and open the door to straightforward Denial of Service (DoS) attacks.

We show that *wormholes* [49], which were so far considered to be a threat, can be used as a reactive defense mechanism: in our solution, thanks to channel diversity, the jammed nodes are able to create a communication route that escapes jamming; thus, appropriate information can be conveyed out of the jammed region. The creation of a wormhole can be triggered by the absence of acknowledgment, after several transmissions. We explain the principle of *probabilistic wormholes* by analyzing three approaches based on this principle. In the first, a network with regular wireless sensor nodes is augmented with a certain number of wired pairs of sensor nodes, therefore resulting in a *hybrid sensor network*. In the second, the deployed nodes (or a subset of them) organize themselves as frequency hopping pairs. For both approaches we compute the probability that at least one wormhole can be formed. Finally, in the third approach, there is no coordination about the communication channel; we analyze this approach through simulations.

The organization of the rest of the chapter is the following. In Section 2.2, we explain the need for the approach based on wormholes. In Section 2.3, we briefly introduce the three solutions that we analyze in this chapter. In Section 2.4, we give a detailed description and we analyze the solution based on wired pairs of sensor nodes. In Section 2.5, we analyze the solution based on frequency

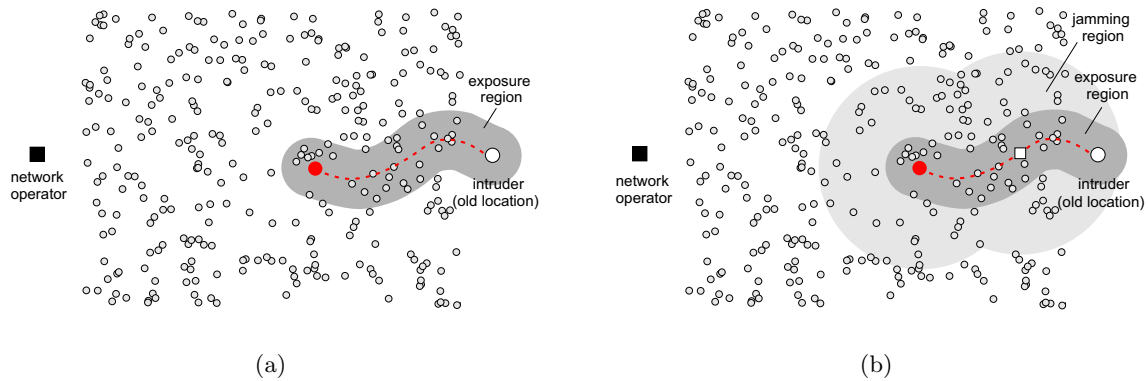


Figure 2.1: The *coverage paradox* – in spite of the fact that an intruder is detected by the sensor nodes (and the network is connected), the network operator cannot be informed on time: (a) The intruder moves in the network and gets detected by the nodes located in the *exposure region*; (b) The intruder moves in the network while stealthily jamming all communication within the *jamming region* (the white square represents a jamming device left behind by the intruder on his way). To avoid detection of jamming by the nodes that do not sense its presence, the intruder can employ a “stealth” jamming strategy.

hopping pairs of sensor nodes. In Section 2.6, we describe and analyze the third solution that is based on uncoordinated channel hopping. We address the related work in Section 2.7. Finally, we summarize the chapter in Section 2.8.

## 2.2 Motivation and Existing Tradeoffs

Our work is motivated by the following scenario. A network of wireless sensors is deployed to detect an event (e.g., the presence of a thief in a museum). Upon detection of the event, a (motion) sensor reports it to the network operator, which then reacts accordingly. Any failure by the sensor to report the event would result in the event being undetected by the operator, and would prevent any action to be taken (in our example, the presence of a thief would be undetected). This failure can occur for several reasons: faulty or compromised sensors, unreliable or disrupted communication links. In this work, we focus on the latter ones.

In a wireless sensor network, all mutual communication between sensors and between sensors and the network operator is wireless (and multi-hop) [9]. This makes it possible for the attacker to jam the communication between sensors and the operator. We show an example of this scenario in Figures 2.1(a) and (b). Figure 2.1(a) shows an intruder (adversary) whose presence is sensed by sensors located within the exposure region (the region from which the adversary’s presence can be sensed). Figure 2.1(b) shows that all communication from the sensors within the exposure region to the rest of the network (to their neighboring sensors) is jammed by the adversary (and an additional jamming device – the white square on the figure), resulting in the presence of the adversary not being reported to the operator (on time). This example shows that an adversary can, by jamming communication between the sensors, effectively *delay* the report about his presence (and, in some cases, prevent being detected at all). Here, we speak about the “delay”, since the sensor nodes from the exposure region may eventually detect the jamming activity of the adversary. However, this is not so easy task considering the computational capabilities of tiny sensor nodes [111]. At the time



some report arrives at the network operator, it may already be too late to take any meaningful action. Note also that the attacker can use some smart jamming strategy, to avoid being detected by the nodes that do not sense its presence (the nodes outside the exposure region - Figure 2.1(b)). Usually, packets in sensor networks have no protection apart from a simple CRC; therefore, only a short jamming pulse is sufficient to destroy a whole packet [82].

Furthermore, even if jamming is detected, the network operator still cannot precisely locate the adversary; only the boundary of the jamming region can be determined (Figure 2.1(b)). Therefore, there is a clear need for defense mechanisms that can ensure *timely data delivery* in spite of jamming attacks.

### 2.2.1 Proactive vs. Reactive Sensor Networks

Generally, we distinguish two basic types of sensor networks: proactive and reactive. Proactive networks involve a periodic flow of data between sensor nodes and the sinks. On the contrary, in reactive networks, packets are sent only when some event of interest occurs and is sensed. Reactive networks are characterized by lower energy consumption and therefore longer network lifetimes.

In the case of proactive sensor networks, several simple solutions can be proposed to ensure that the operator receives event reports or detects jamming. One solution consists in having sensors periodically report their status to the network operator (e.g., upon query from the operator); if a sensor does not report its status within an expected period, the operator can request a re-transmission or conclude that the communication from that sensor is prevented by an adversary. If these status reports are sent very frequently, sensor batteries will be exhausted in a short time; if they are sent infrequently, the batteries will last longer, but the time elapsed between an event happened and its reporting can be long and might render the alarm useless. Another similar solution is that sensors hold the list of their neighbors and periodically poll them to check if the communication links between them are still valid. This solution has similar drawbacks as the first proposal, as it either has high energy cost (if the polls are frequent), or opens a time window within which an event is undetected (if the polls are not frequent).

These and similar proactive solutions require the sensors to periodically communicate even if no event has occurred. Furthermore, these solutions do not ensure that the network operator is informed about the event immediately after it happens. We therefore argue that instead of being proactive, in many applications event reporting need to be reactive, saving energy (as the sensors communicate only when an event is detected) and enabling the network operator to be informed about an event within a reasonably short time period.

Reactive event reporting is, however, vulnerable to jamming, because if the communication from a sensor to the operator is jammed, the operator will not raise any alarm as it does not expect any reports to come at any given time. It is therefore important to ensure that, if a sensor detects an event, it can communicate this event to the network operator despite adversary's jamming.

In this chapter, we will show how to build a reactive sensor network that guarantees timely delivery of event reports from the sensors to network authorities in the presence of an adversary that tries to remain undetected.

### 2.2.2 Straightforward Solutions Might Not Be Adequate

We describe now an example of a straightforward solution. When a node senses the presence of the intruder, it begins to jam its neighbors (i.e., it transmits and disables the carrier sense and the

potential random backoff procedure). If this is done on each node, the jamming activity will spread throughout the network and eventually it will reach the nodes that are close to some sink.

Clearly, with this approach even the most naive attacker becomes a “nightmare” for the network operator: a strictly local (involving a single sensor node) jamming attack can disable the whole network. In addition, the risk of false alarms is very high. Therefore, when designing a security solution for this (and any) kind of networks, a special care must be taken to avoid potential undesirable secondary consequences.

## 2.3 Proposed Solution: Probabilistic Wormholes

In the following three sections, we present and analyze three mechanisms to achieve timely event reporting

- *Wired pairs of sensor nodes;*
- *Coordinated frequency-hopping pairs;*
- *Uncoordinated channel-hopping.*

Here we give just a high level overview of the proposed approach. In our solution, a portion of pairs of sensor nodes create (probabilistically) communication links that are resistant to jamming. By not requiring all the sensor nodes in the network to have this capability, we actually trade-off the network robustness with the network complexity (and the cost). Now, for the given randomly located adversary (attacker), there will be a positive probability that a sensor node, residing in the exposure region of the attacker, forms a (multihop) path from the exposure region to the region not affected by jamming, in such a way that this path is not affected by ongoing jamming. We call such a path the *probabilistic wormhole*. An example of a probabilistic wormhole, realized through wires, is shown on Figure 2.2(b).

We emphasize here that our goal is not to propose a single solution but rather to explain and motivate the principle of *probabilistic wormholes*.

## 2.4 Wormholes via Wired Pairs of Sensor Nodes

In this solution, we propose to augment a wireless sensor network with a certain number of pairs of sensor nodes that are each connected through a wire. Connected sensor nodes are also equipped with wireless transceivers, just like regular sensor nodes. As a result we obtain a hybrid sensor network as shown on Figure 2.2(a): isolated points represent regular nodes and connected pairs are denoted as connected points. A similar form of a hybrid sensor network already appears in the context of the NIMS project [55], and in the work by Sharma and Mazumdar [96].

### 2.4.1 Rationale of Wired Pairs

We now introduce some terminology and explain the operating principles underlying the approach based on wired pairs of sensor nodes. We denote with  $d$  the length of the wire connecting a pair of nodes; we assume all pairs to be connected with wires of the same length. Assuming random deployment of connected pairs (e.g., by throwing them from an aircraft), the distance between the nodes of a given connected pair, once the pair lands in the field, is a random variable taking values from interval  $[0, d]$ . We further denote with  $R_t$  the transmission range of the wireless transceivers

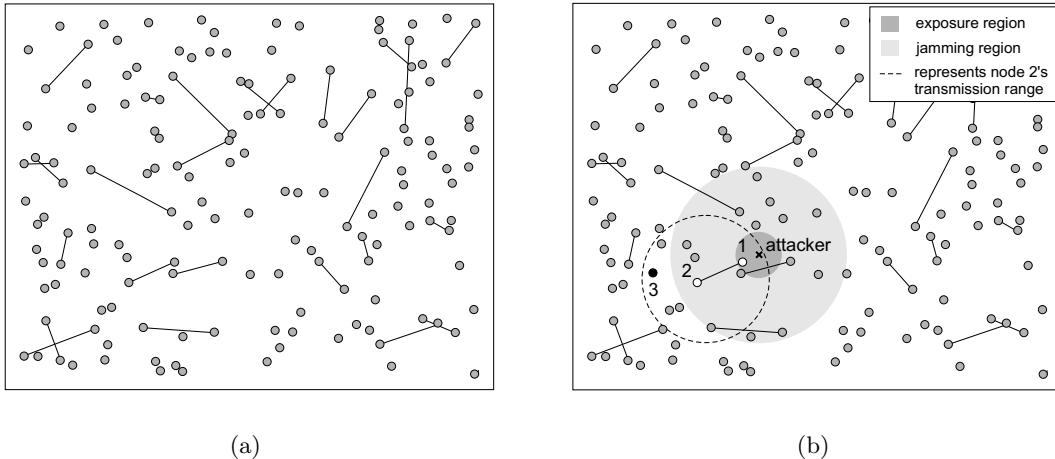


Figure 2.2: Probabilistic wormholes via wired pairs of sensor nodes: (a) Hybrid sensor network with randomly deployed sensor nodes: isolated points are regular nodes, connected points represent sensor nodes connected through a wire.; (b) Hybrid sensor network with an attacker who jams the surrounding nodes. Connected pair (1, 2) and regular node 3 create a *wormhole* from the exposure region to the region that is not jammed.

mounted on the sensor nodes. Let us now consider the scenario shown on Figure 2.2(b). In this scenario, the attacker ( $A$ ), represented by sign  $x$ , stealthily jams the region (called *jamming region*) within jamming range  $R_j$ . We call the *exposure region* the region that surrounds the attacker and from which the attacker's presence can be detected. As can be seen on Figure 2.2(b), we model the exposure region by a circle centered at the location of the attacker. We denote with  $R_s$  the radius of the exposure region. The exposure region is related to the sensing capabilities of the employed sensors, which is the reason for using subscript  $s$  in  $R_s$ . Note, however, that the notion of exposure region is much broader. For example, when the attacker jams some area, the nodes whose transmissions are affected by this attack can deduce that an attack is taking place by observing multiple failures to receive the ACK from their intended destinations. In this case, all such nodes make the exposure region.

In order to prevent any report (e.g., a report about the attacker's presence), generated by the regular nodes located within the exposure region, to successfully leave the exposure region, the attacker simply jams the area within jamming range  $R_j \geq R_t + R_s$ . In this situation, the connected pairs serve as a rescue. In our example on Figure 2.2(b), connected pair (1, 2) creates a link resistant to jamming from the exposure region. When node 1 senses the presence of the attacker, it makes use of the wired channel to communicate a short report to its peer node 2. Since the wired channel between nodes 1 and 2 is not affected by the jamming activity of the attacker, the report sent by node 1 is successfully received by node 2. In turn, node 2 simply transmits (broadcasts) this report using the wireless transceiver with transmission range  $R_t$ . Some node (e.g., node 3 on Figure 2.2)(b) that is located within transmission range  $R_t$  from node 2 and outside of the jamming region, will potentially receive the report and pass it further, possibly over multiple hops, to some sink. Therefore, the 2-hop path between nodes 1 and 3 can be thought of as a *wormhole* that is resistant to the ongoing jamming activity by the attacker.

Of course, the attacker can simply increase the jamming region in such a way that the attacker

also jams node 3. However, in the same way, the network operator can further increase the transmission range ( $R_t$ ) of the wireless transceivers, the length of the wire ( $d$ ), as well as the exposure region (by deploying more advanced sensors with more advanced sensing capabilities). In addition, if a jamming signal is stronger, the probability that it gets detected and reported increases. In the following section, we develop an approximation model that allows us to better understand potential benefits of changing the system parameters:  $R_t$ ,  $R_s$ ,  $d$  and  $R_j$ , as well as the node density.

There are many technical issues to address within the approach proposed in this section. In this work, however, our goal is to establish the relationship between the probability that at least one wormhole (from the given exposure region) is created and different system parameters.

## 2.4.2 Performance Analysis

We assume the regular sensor nodes to be deployed randomly with uniform distribution in the deployment region  $\mathcal{D}$ . The deployment region  $\mathcal{D}$  is modelled by a  $D \times D$  square,  $D < \infty$  (see Figure 2.3(a)). We denote with  $n$  the number of regular nodes deployed in  $\mathcal{D}$ . We further approximate exposure and jamming regions with circles of radius  $R_s$  and  $R_j$ , respectively (the Boolean model). Finally, we assume that the jamming range satisfies  $R_j \geq R_s + R_t$ . The center point  $(x_A, y_A) \in \mathcal{D}$  of the exposure (jamming) region represents the location of the attacker. In our model, we assume both exposure and jamming regions to be contained completely within the deployment region; this is to avoid cumbersome technicalities with boundary regions (Figure 2.3(a)). For convenience we set  $(x_A, y_A) = (0, 0)$  (Figure 2.3(a)). We also assume that the attacker is ignorant of the locations of connected pairs<sup>1</sup>; in other words, the attacker's location is assumed to be independent of the locations of the connected pairs.

To model the random deployment of connected pairs we proceed as follows. Let us consider connected pair (4, 5) on Figure 2.3(b). We first pick a point  $(x_{4,5}, y_{4,5})$  uniformly at random from  $\mathcal{D}$ . Next, we draw (or, rather, imagine) a *deployment disk* of radius  $d/2$  around the point  $(x_{4,5}, y_{4,5})$  (Figure 2.3(b)). Finally, we pick two points  $(x_4, y_4)$  and  $(x_5, y_5)$ , uniformly at random and independently, from the area enclosed by the deployment disk centered at  $(x_{4,5}, y_{4,5})$ ;  $(x_4, y_4)$  and  $(x_5, y_5)$  then correspond to the positions of connected nodes 4 and 5, respectively (Figure 2.3(b)). Note that the deployment disk (with diameter  $d$ ) ensures that the link (wire) between nodes 4 and 5 does not exceed the maximum length of  $d$ . This procedure is then repeated (independently) for each of the  $K$  connected pairs to be deployed.

More formally, with each connected pair  $(i, j)$  to be deployed in the deployment region  $\mathcal{D}$ , we can associate three 2-dimensional random variables:  $\mathbf{P}_{i,j} = (X_{i,j}, Y_{i,j})$ ,  $\mathbf{P}_i = (X_i, Y_i)$  and  $\mathbf{P}_j = (X_j, Y_j)$ , where  $X_{i,j} \in [0, D]$  and  $Y_{i,j} \in [0, D]$  are uniform (continuous) random variables, and  $(X_i, Y_i)$  and  $(X_j, Y_j)$  are (jointly continuous) uniform random variables taking values from the set  $\{(x, y) : (x - x_{i,j})^2 + (y - y_{i,j})^2 \leq (d/2)^2, \text{ for fixed } (x_{i,j}, y_{i,j}) \in \mathcal{D}\}$ . Thus, for the given connected pair  $(i, j)$ ,  $\mathbf{P}_{i,j}$  describes the location of the center point of the corresponding deployment disk, while  $\mathbf{P}_i$  and  $\mathbf{P}_j$  describe the locations of nodes  $i$  and  $j$ , respectively.

For the given attacker, located at point  $(x_A, y_A) = (0, 0)$ , we want to calculate the probability that at least one wormhole exists from the corresponding exposure region into the region not affected by the attacker's jamming activity. For example, on Figure 2.3(b), nodes 1, 2 and 3 form such a wormhole. We denote with

$$P[\text{at least one wormhole} | (x_A, y_A)]$$

---

<sup>1</sup>This assumption is more legitimate in the context of the solution based on frequency-hopping pairs (studied in Section 2.5). Note, however, that information about the locations of connected pairs becomes less relevant as the density of the connected pairs increases.

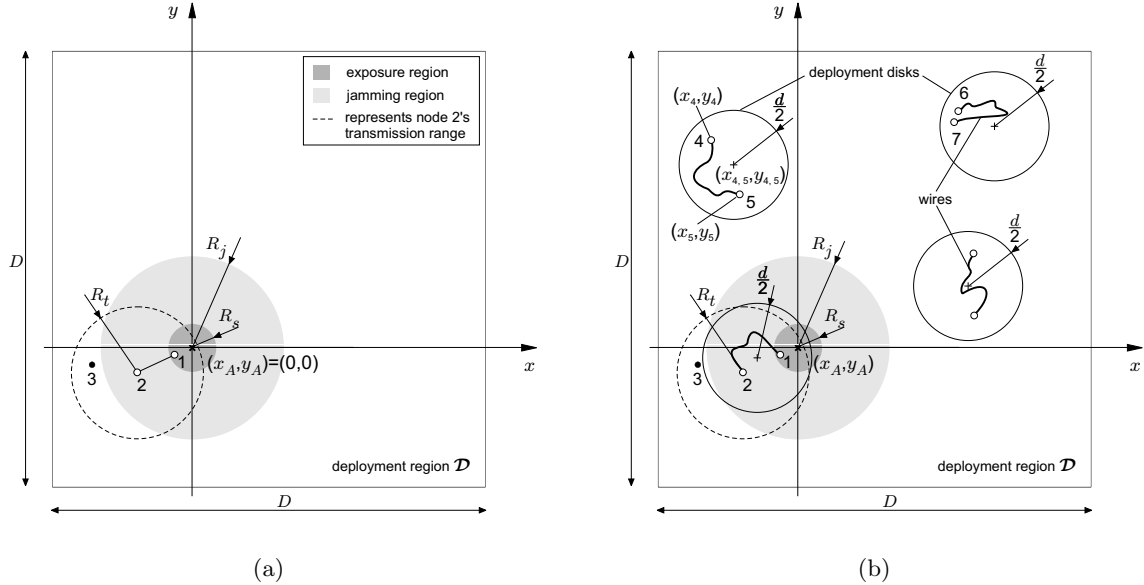


Figure 2.3: (a) Geometry used in the analysis of the solution based on probabilistic wormholes; (b) Approximation model for random deployment of connected pairs (the thick curves connecting the nodes represent wires between the nodes).

the above probability.

Let us consider a single connected pair  $(k, l)$ . To calculate  $P[\text{at least one wormhole} | (x_A, y_A)]$ , we first define the following event

$S \stackrel{def}{=} \{ \text{the connected pair } (k, l) \text{ forms a wormhole from the exposure region around } (x_A, y_A) \text{ to the area not affected by jamming} \}$ .

It is important to stress here that we require a wormhole to always involve at least one regular node, even in cases when the connected pair itself is sufficient to form a wormhole from the jamming region (for example, this may happen when  $d > R_s + R_j$ ).

Let  $P[S]$  be the probability of event  $S$  and let  $p_s$  denote the value of  $P[S]$ . By assumption: (1) the location of any connected pair  $(i, j)$  is independent of the attacker's position  $(x_A, y_A)$ , and (2) the positions of the connected pairs are sampled from the same distributions and independently. Therefore,  $p_s$  is equal for all the deployed connected pairs. Since there are  $K$  connected pairs deployed randomly and independently, we finally obtain the following:

$$\begin{aligned} P[\text{at least one wormhole} | (x_A, y_A)] &= 1 - (1 - p_s)^K \\ &\approx 1 - e^{-Kp_s} \end{aligned} \quad (2.1)$$

where the last approximation is valid for small  $p_s$  and large  $K$ . We now calculate  $p_s = P[S]$ . From the definition of the random variable  $\mathbf{P}_{k,l} = (X_{k,l}, Y_{k,l})$ , we know that its probability density function satisfies  $f_{\mathbf{P}_{k,l}}(x, y) = f_{X_{k,l}, Y_{k,l}}(x, y) = 1/D^2$ . Then, by the law of total probability we can write for  $P[S]$ :

$$P[S] = \iint_{(x,y) \in \mathcal{D}} P[S | \mathbf{P}_{k,l} = (x, y)] f_{\mathbf{P}_{k,l}}(x, y) dx dy \quad (2.2)$$

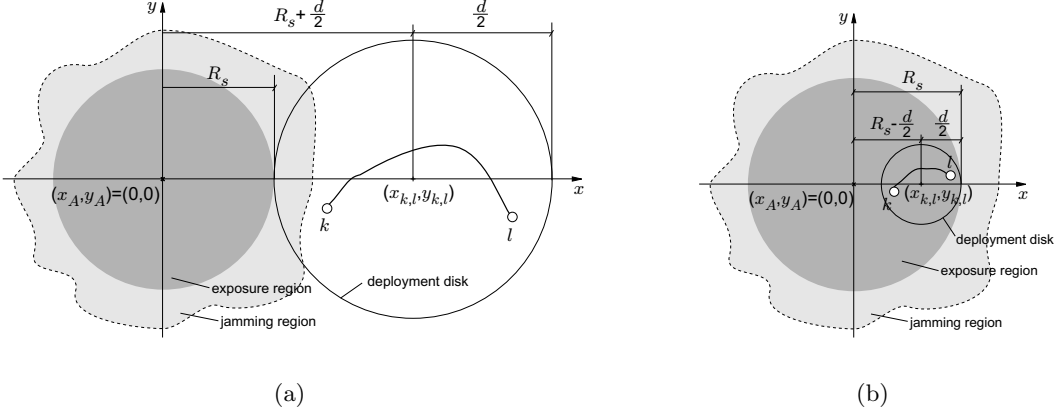


Figure 2.4: Examples where connected pair  $(k, l)$  cannot create a wormhole (note that only a part of the jamming region is shown): (a) An example where connected pair  $(k, l)$  cannot create a wormhole with  $R_s < d/2$ ; (b) An example where connected pair  $(k, l)$  cannot create a wormhole with  $R_s > d/2$ .

Observe now that for many points  $(x, y) \in \mathcal{D}$ , we will have  $P[S|\mathbf{P}_{k,l} = (x, y)] = 0$ . For example,  $P[S|\mathbf{P}_{k,l} = (x, y)] = 0$  for all points  $(x, y)$  that happen to be located far enough from  $(x_A, y_A) = (0, 0)$ , that is, points for which  $\text{dist}\{(x, y), (0, 0)\} > R_s + d/2$ , where  $\text{dist}\{(x, y), (0, 0)\}$  is the Euclidian distance between points  $(x, y)$  and  $(0, 0)$  (see Figure 2.4(a)). Likewise, for  $d/2 < R_s$ , if  $\text{dist}\{(x, y), (0, 0)\} < R_s - d/2$ , then  $P[S|\mathbf{P}_{k,l} = (x, y)] = 0$  as well (see Figure 2.4(b)); in this case, since  $R_j \geq R_t + R_s$ , neither node  $k$  nor node  $l$  can reach any regular node that is located outside of the jamming region. Therefore, using the polar coordinates

$$\begin{aligned} (x, y) &= (r \cos \theta, r \sin \theta) \\ r &= \text{dist}\{(x, y), (0, 0)\} , \end{aligned}$$

expression (2.2) can be rewritten as follows

$$P[S] = \frac{1}{D^2} \iint_{\substack{r \in [\underline{r}, R_s + \frac{d}{2}] \\ \theta \in [0, 2\pi]}} P[S|\mathbf{P}_{k,l} = (r \cos \theta, r \sin \theta)] r dr d\theta , \quad (2.3)$$

where  $\underline{r} = R_s - \frac{d}{2}$  if  $\frac{d}{2} \leq R_s$  and  $\underline{r} = 0$  if  $\frac{d}{2} \geq R_s$ . For notational simplicity, in the sequel, we will use  $P[S|\mathbf{P}_{k,l} = (r, \theta)]$  as the shorthand for  $P[S|\mathbf{P}_{k,l} = (r \cos \theta, r \sin \theta)]$ .

We next calculate  $P[S|\mathbf{P}_{k,l} = (r, \theta)]$ , to be able to calculate  $P[S]$  from expression (2.3). For this we need some additional notation. We first define the following event

$$W_1 \equiv \left\{ \text{one node of the connected pair } (k, l) \text{ is located within the exposure region and the other outside of the exposure region} \right\} .$$

For example, for pair  $(k, l) = (1, 2)$  on Figure 2.3(b), event  $W_1$  has occurred. Furthermore, we define the following event:

$$W_2 \equiv \left\{ \text{for the connected pair } (k, l) \text{ there exists at least one regular node that is located outside of the jamming region but within the transmission range } R_t \text{ of either } k \text{ or } l \right\} .$$

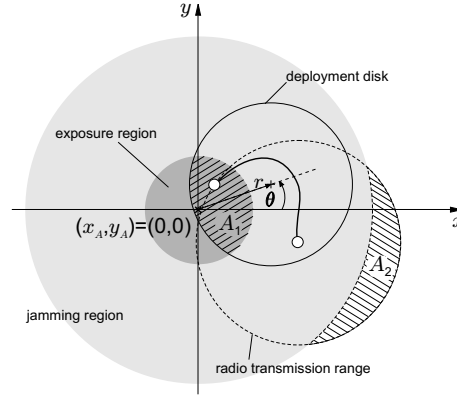


Figure 2.5: Definition of regions  $A_1(r, \theta)$  and  $A_2$ .

For example, for pair  $(k, l) = (1, 2)$  on Figure 2.3(b), event  $W_2$  has occurred, since node 2 has regular node 3 that is located within node 2's radio transmission range and outside of the jamming range.

**Fact 1** *Assume  $R_j \geq R_t + R_s$ . Then the following is true:  $S \equiv W_1 \wedge W_2$ , i.e., event  $S$  happens if and only if both event  $W_1$  and event  $W_2$  happen.*

Assume that  $W_1$  has happened. In this case, assume (without any loss of generality) that node  $k$  is located within the exposure region and node  $l$  outside of it. Since  $R_j \geq R_t + R_s$ , all the regular nodes that are located within node  $k$ 's transmission range  $R_t$  must also fall in the jamming range. Since  $W_2$  happens as well, there must be at least one regular node  $m$  that is located within the transmission range of node  $l$  and outside of the jamming region. But then nodes  $k, l$  and  $m$  form a wormhole from the exposure region, i.e., event  $S$  has happened. If one among  $W_1$  and  $W_2$  does not happen, neither does  $S$ .

From Fact 1, we have the following:

$$\begin{aligned} P[S|\mathbf{P}_{k,l} = (r, \theta)] &= P[W_1, W_2|\mathbf{P}_{k,l} = (r, \theta)] \\ &= P[W_1|\mathbf{P}_{k,l} = (r, \theta)]P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)]. \end{aligned} \quad (2.4)$$

Since the positions of peer nodes  $k$  and  $l$  are chosen randomly and independently in the corresponding deployment disk (of radius  $d/2$ ) centered at  $(x, y) = (r \cos \theta, r \sin \theta)$ , we have:

$$P[W_1|\mathbf{P}_{k,l} = (r, \theta)] = 2 \times \frac{|A_1(r, \theta)|}{(d/2)^2 \pi} \times \frac{(d/2)^2 \pi - |A_1(r, \theta)|}{(d/2)^2 \pi}, \quad (2.5)$$

where  $A_1(r, \theta)$  is the set of points  $(x, y) \in \mathcal{D}$  that are located in the *intersection region* obtained as the intersection between the deployment disk (of the pair  $(k, l)$ ) centered at  $(x, y) = (r \cos \theta, r \sin \theta)$  and the exposure region (see Figure 2.5), and  $|A_1(r, \theta)|$  denotes the area (not the set size) of this intersection region.

From Figure 2.5 we can observe that  $|A_1(r, \theta)| = |A_1(r)|$ , i.e., the area  $|A_1(r, \theta)|$  does not depend on  $\theta$ ; note that this is the consequence of setting  $(x_A, y_A) = (0, 0)$  and our assumption that jamming and exposure regions are contained completely within the deployment area<sup>2</sup>. The value of  $|A_1(r)|$  can be computed by the well known formula for the area of circle-to-circle intersection.

<sup>2</sup>By relaxing this assumption, intersection areas  $A_1$  take more complex forms, which significantly increases the complexity of their evaluation.

Next, we evaluate the conditional probability  $P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)]$ . Since event  $W_1$  has happened, it means that one node from the observed pair  $(k, l)$  resides in the exposure region (say node  $k$ ) and the other one (node  $l$ ) is located outside of the exposure region. As we argued right after Fact 1, this implies that  $k$  has no neighbors among regular nodes that are located outside of the jamming region. Then, the event  $W_2$  conditioned on  $W_1$  (which we denote with  $\tilde{W}_2$ ) actually reads

$$\tilde{W}_2 \equiv \left\{ \text{node } l \text{ has at least one neighboring regular node that is located outside of the jamming region} \right\} .$$

Therefore,

$$P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)] = P[\tilde{W}_2|\mathbf{P}_{k,l} = (r, \theta)] . \quad (2.6)$$

Let us denote with  $Disk_{k,l}(r, \theta)$  the set of all the points from the pair  $(k, l)$ 's deployment disk, centered at  $(x, y) = (r \cos \theta, r \sin \theta)$  (see Figure 2.5). Then, by the law of total probability we have:

$$P[\tilde{W}_2|\mathbf{P}_{k,l} = (r, \theta)] = \iint_{(x,y) \in \bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] f_{\mathbf{P}_l}(x, y) dx dy , \quad (2.7)$$

where  $\bar{A}_1(r, \theta) = Disk_{k,l}(r, \theta) - A_1(r, \theta)$ ,  $\mathbf{P}_l$  is the 2-dimensional random variable describing the location of node  $l$ , and  $f_{\mathbf{P}_l}(x, y)$  is the probability density function of the location of node  $l$ , that is,

$$f_{\mathbf{P}_l}(x, y) = \frac{1}{|\bar{A}_1(r, \theta)|} = \frac{1}{(d/2)^2 \pi - |A_1(r)|} \stackrel{def}{=} f_{\mathbf{P}_l}(r) . \quad (2.8)$$

Recall,  $|A_1(r, \theta)| = |A_1(r)|$  (see Figure 2.5).

Since the regular nodes are deployed uniformly at random in  $\mathcal{D}$ , we have for  $(x, y) \in \bar{A}_1(r, \theta)$ :

$$\begin{aligned} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] &= 1 - \left( 1 - \frac{|A_2(x, y)|}{D^2} \right)^n \\ &\approx 1 - e^{-n|A_2(x, y)|/D^2} , \end{aligned} \quad (2.9)$$

where  $A_2(x, y)$  is the set of points from the node  $l$ 's transmission region, which does not fall in the jamming region (see Figure 2.5),  $|A_2(x, y)|$  is the area of this region, and  $n$  is the number of regular nodes deployed. Note that the approximation in expression (2.9) is valid for large  $n$  and  $|A_2(x, y)| \ll D^2$ .

Now, by combining expressions (2.4)-(2.9), we can calculate  $P[S|\mathbf{P}_{k,l} = (r, \theta)]$  as follows

$$\begin{aligned} P[S|\mathbf{P}_{k,l} = (r, \theta)] &\stackrel{(1)}{=} P[W_1|\mathbf{P}_{k,l} = (r, \theta)] P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)] \\ &\stackrel{(2)}{=} P[W_1|\mathbf{P}_{k,l} = (r, \theta)] P[\tilde{W}_2|\mathbf{P}_{k,l} = (r, \theta)] \\ &\stackrel{(3)}{=} P[W_1|\mathbf{P}_{k,l} = (r, \theta)] \iint_{(x,y) \in \bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] f_{\mathbf{P}_l}(x, y) dx dy \\ &\stackrel{(4)}{=} P[W_1|\mathbf{P}_{k,l} = (r, \theta)] f_{\mathbf{P}_l}(r) \iint_{(x,y) \in \bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] dx dy \\ &\stackrel{(5)}{=} 2 \times \frac{|A_1(r)|}{(d/2)^2 \pi} \times \frac{(d/2)^2 \pi - |A_1(r)|}{(d/2)^2 \pi} \times \frac{1}{(d/2)^2 \pi - |A_1(r)|} \\ &\quad \times \iint_{(x,y) \in \bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] dx dy \\ &\stackrel{(6)}{\approx} \frac{32|A_1(r)|}{(d^2 \pi)^2} \iint_{(x,y) \in \bar{A}_1(r,\theta)} \left( 1 - e^{-\frac{n|A_2(x,y)|}{D^2}} \right) dx dy , \end{aligned} \quad (2.10)$$



where (1) follows from the expression (2.4), (2) follows from the expression (2.6), (3) follows from (2.7), (4) follows from the fact that for fixed  $r$  the probability density function  $f_{\mathbf{P}_1}(r)$  is a constant (see the expression (2.8)), (5) follows from the expressions (2.5) and (2.8) and the fact that the area  $|A_1(r)|$  is independent of  $\theta$ , and finally (6) follows from the approximation in the expression (2.9).

Finally, by plugging the expression (2.10) in the expression (2.3) we obtain

$$P[S] \approx \frac{1}{D^2} \iint_{\substack{r \in [r, R_s + \frac{d}{2}] \\ \theta \in [0, 2\pi]}} \left\{ \frac{32|A_1(r)|}{(d^2\pi)^2} \iint_{(x,y) \in \bar{A}_1(r,\theta)} \left( 1 - e^{-\frac{n|A_2(x,y)|}{D^2}} \right) dx dy \right\} r dr d\theta \quad (2.11)$$

$$\stackrel{(1)}{=} \frac{64}{D^2 d^4 \pi} \int_{r \in [r, R_s + \frac{d}{2}]} \left\{ \iint_{(x,y) \in \bar{A}_1(r)} \left( 1 - e^{-\frac{n|A_2(x,y)|}{D^2}} \right) dx dy \right\} |A_1(r)| r dr ,$$

where (1) follows by observing that  $|A_2(x,y)|$  (and therefore  $\{1 - \exp(-n|A_2(x,y)|/D^2)\}$ ) is independent of  $\theta$  (see Figure 2.5).

Due to the complex expressions for areas  $|A_1(r)|$  and  $|A_2(x,y)|$ , integrating analytically the resulting expression for  $P[S]$  is very hard. For this reason, in Section 2.4.3 we solve the expression (2.11) numerically and validate it by simulations.

Assume now that we want to achieve  $P[\text{at least one wormhole}|(x_A, y_A)] \geq p_w$ , where  $p_w$  is some targeted probability. Let  $K_0$  denote the critical (minimum) number of connected pairs for which  $P[\text{at least one wormhole}|(x_A, y_A)] = p_w$  holds. Then, from (2.1) we have the following result.

**Theorem 8**

$$K_0 = \frac{\ln(1 - p_w)}{\ln(1 - p_s)} \approx -\frac{\ln(1 - p_w)}{p_s} , \quad (2.12)$$

where  $p_s$  is given by the expression (2.11).

Note that, due to our assumption about independence between the deployment of different connected pairs and their independence of the attacker's location,  $p_s$  is the probability that an arbitrary connected pair forms a wormhole. The result from Theorem 8 is common in stochastic geometry.

### 2.4.3 Simulations and Model Validation

We investigated the proposed analytical model by means of simulations. We evaluated  $P[\text{at least one wormhole}|(x_A, y_A)]$  as a function of parameters  $K, R_s, n$  and  $d$ . In our simulations we set  $R_j = R_s + R_t$ . For each parameter, we perform 20 experiments as follows. For each different value of a given parameter (i.e.,  $R_s, K, n, d$ ), we first generate randomly the network topology with  $n$  regular nodes and  $K$  connected pairs (see Figure 2.2(a)). Next, we throw randomly  $N = 500$  jamming regions (circles of radius  $R_j$ ) in the deployment area of size  $D \times D$ . Then we count the number  $n_W \leq N$  of jamming regions for which there is at least one wormhole. From this we calculate the relative frequency  $f_W(N) = n_W/N$ . Finally, we average the results obtained from 20 experiments and present them with 95% confidence interval.

The results are shown on Figure 2.6 and Figure 2.7, together with numerical results obtained from the analytical model developed in the previous section. As we can see from the figures, the analytical model predicts quite accurately the probability that at least one wormhole is created. Other interesting conclusions can be drawn from the figures. We can see that the increase in either  $R_s$  and  $K$  results in nearly linear increase in  $P[\text{at least one wormhole}|(x_A, y_A)]$ . We can further see that the best ‘‘investment’’ for the network operator is to increase the size of

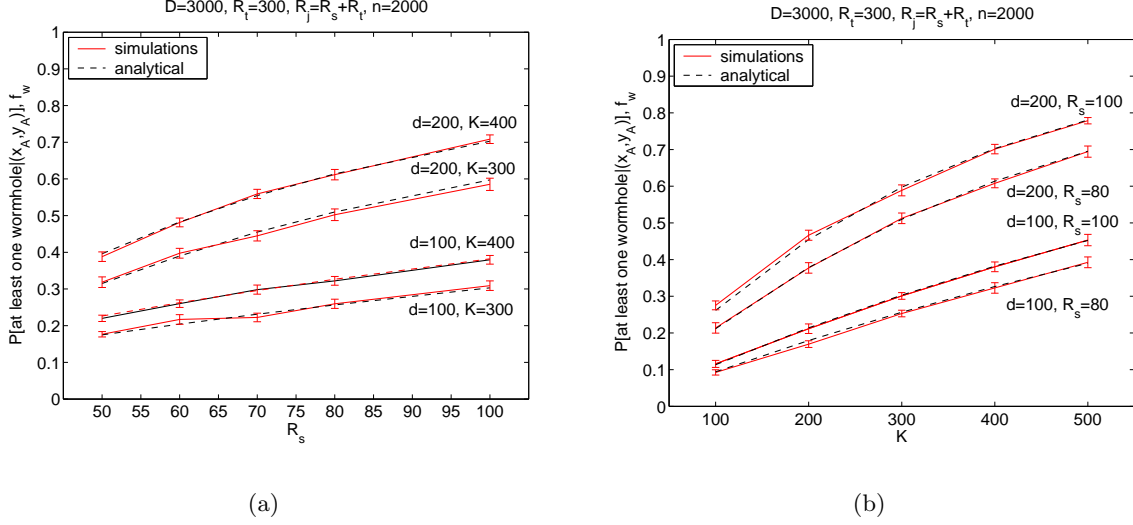


Figure 2.6:  $P[\text{at least one wormhole}|(x_A, y_A)]$  and relative frequency  $f_W(500)$  vs. (a) the size of the exposure region  $R_s$  and (95% confidence interval) and (b) the number of connected pairs  $K$ . We use 95% confidence interval.

the exposure region (e.g., by using more advanced sensing mechanisms). For example, an increase of  $R_s$  for 20 units (from 80 to 100), for  $K = 300$  and  $d = 200$ , results in the increase of  $P[\text{at least one wormhole}|(x_A, y_A)]$  of around 0.1 (Figure 2.6(a)). However, an increase of  $K$  for 100 units (300 to 400), for  $d = 200$  and  $R_s = 100$ , results in nearly the same increase of  $P[\text{at least one wormhole}|(x_A, y_A)]$ , i.e., around 0.12 (Figure 2.6(b)). Therefore, we can trade-off the number of wired pairs required with the size of the exposure region (for example, by using more advanced sensing technology). The advantage of increasing  $R_s$  versus  $K$  can easily be seen by taking the first derivative of  $P_w \stackrel{\text{def}}{=} P[\text{at least one wormhole}|(x_A, y_A)]$  with respect to  $p_s$  and  $K$ . From the expression (2.1)

$$\frac{\partial P_w}{\partial p_s} = K e^{-K p_s} \quad \text{and} \quad \frac{\partial P_w}{\partial K} = p_s e^{-K p_s} .$$

Since  $p_s$  increases in  $R_s$ , it follows readily that it is more advantageous to increase  $R_s$  than  $K$ . From Figure 2.6(a) and Figure 2.6(b) we can further see that the cable length plays a major role; we note, however, that this is partially because we take  $R_j = R_t + R_s$ .

From Figure 2.7(a) and Figure 2.7(b) we observe that increasing  $n$  and  $d$  is beneficial only until a certain saturation point. As far as  $n$  is concerned, at the saturation point, we have  $P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)] \approx 1$  (expression (2.9)), and hence  $P[S|\mathbf{P}_{k,l} = (r, \theta)] \approx P[W_1|\mathbf{P}_{k,l} = (r, \theta)]$ , where probability  $P[W_1|\mathbf{P}_{k,l} = (r, \theta)]$  is not a function of  $n$  (see expression (2.5)). Note that the average distances between connected peers are significantly shorter than the maximum length  $d$ ; the average distance between two connected nodes is around  $0.45 \times d$  (which is consistent with the expected distance between two randomly selected points from a disk of radius  $d/2$  [98]).

The results from this section show that while feasible, the solution based on pairs of nodes connected through wires is expensive in terms of the number of wires needed and their length. In the following section, we propose and analyze an alternative and “light” approach to creating wormholes.

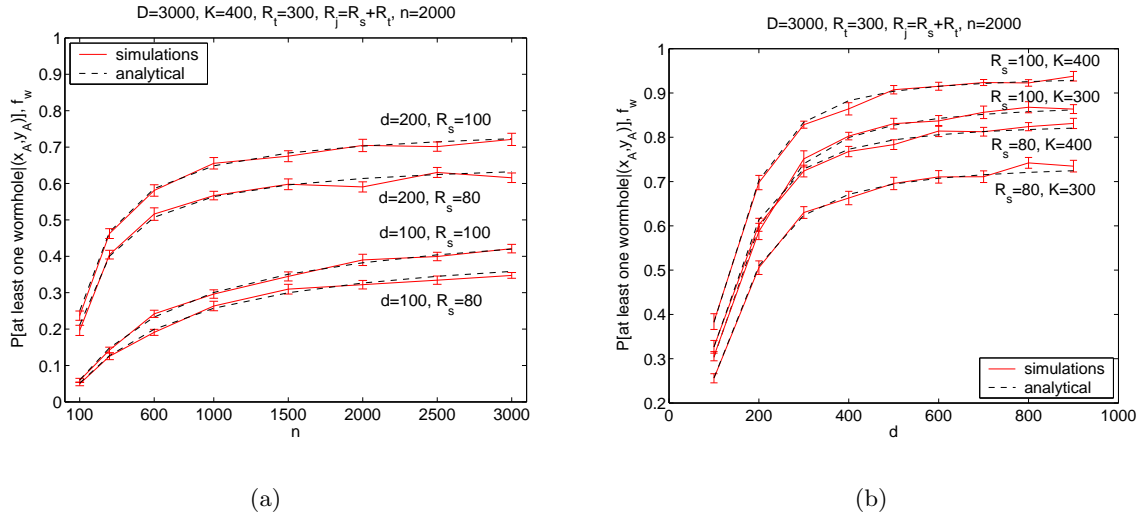


Figure 2.7:  $P[\text{at least one wormhole}|(x_A, y_A)]$  and relative frequency  $f_w(500)$  vs. (a) the number of regular nodes  $n$ , and (b) the maximum wire length  $d$ . We use 95% confidence interval.

We also conclude that the analytical model developed in Section 2.4.2 provides a solid ground for the understanding of important trade-offs in the solution based on connected pairs.

## 2.5 Wormholes via Coordinated Frequency Hopping Pairs

The solution based on pairs of nodes connected through wires has the major drawback that it requires the wires to be deployed in the field. Moreover, as we saw in Section 2.4.3, in order to achieve a reasonably high  $P[\text{at least one wormhole}|(x_A, y_A)]$ , the number of connected pairs (and therefore wires) to be deployed can be very high. In this section, we propose a solution similar to the previous one, with the only difference that the pairs are formed exclusively through wireless links resistant to jamming. By using a wireless link, not only do we avoid cumbersome wires, we can also afford longer links between pairs<sup>3</sup>; as we saw in Section 2.4.3 (Figure 2.7(b)), the increase in  $d$  (maximum length of a wire) has a profound impact on  $P[\text{at least one wormhole}|(x_A, y_A)]$ .

### 2.5.1 Rationale of Frequency Hopping (FH) Pairs

In the solution based on coordinated frequency hopping pairs, we distinguish two types of sensor nodes. The first type are *regular nodes* equipped with an ordinary single-channel radio. The second type are sensor nodes equipped with two radios: the regular radio and a radio with frequency-hopping (FH) capability (e.g., Bluetooth). We note that there already exist several sensor platforms having FH capabilities [1]. It is important to stress, however, that we do not propose to equip all the nodes in the network with FH radio (a case study of Bluetooth sensor networks can be found in [69]). The reason is that FH radio imposes a substantial overhead on sensor nodes in multihop networks [69]; the need for “synchronization” (at multiple levels) between senders and designated

<sup>3</sup>We note, however, that a few wired links could still be used as a means to reduce the energy consumption of nodes, by creating shortcuts between different parts of the network.

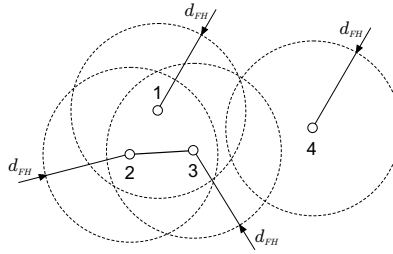


Figure 2.8: On-line FH pairing process: the thick line connecting FH nodes 2 and 3 means that they form a FH pair, while FH nodes 1 and 4 remain “unpaired” ( $d_{FH}$  is the radio transmission range of the FH nodes).

receivers (synchronization of hopping sequences, time synchronization) might be a major deterrent to using FH radios in multihop wireless sensor networks [69].

Instead, we propose to deploy a certain number of FH enabled nodes along with the regular nodes. We assume that the attacker cannot jam the employed FH radio. Once deployed (in the bootstrapping phase; no attack takes place yet), each FH enabled node begins to look for another FH node among its FH neighbors. Once two FH neighboring nodes agree to form a FH pair, they generate a random frequency-hopping sequence (which is ideally unique in the 2-hop neighborhood of a given pair). In this work, we restrict each FH node to be member of at most one FH pair. We denote with  $d_{FH}$  the transmission range of the FH radio (i.e., FH nodes), where  $d_{FH}$  may be different from the transmission range  $R_t$  of regular nodes (radio).

The solution based on FH pairs is similar to the previous one based on wired wormholes. Here again, our goal is to ensure that FH pairs form at least one wormhole, with a high probability, in the event of a jamming attack (see Figure 2.2(b)). The important difference with respect to the solution based on wires is that the formation of FH pairs takes place once the nodes are deployed in the field - the *opportunistic pairing process*. FH hopping enabled nodes will use some form of a *pairing protocol* to discover their FH enabled neighbors and to eventually form a pair with one of them. A simple opportunistic pairing protocol would be to let every node advertise its availability until it makes a FH pair with a randomly selected “available” node or it fails to find some “free” (available) neighbor. The details of such a pairing protocol are out of the scope of this work. We, however, expect it to be probabilistic in nature<sup>4</sup> (for example, due to the probabilistic channel access mechanisms). For this reason (and because of the random deployment of FH enabled nodes), it is very likely that some FH nodes will not find any “free” FH neighbor.

Consider the example on Figure 2.8, where FH nodes 1, 2 and 3 are all neighbors to each other (i.e., they are located within  $d_{FH}$  of each other) and FH node 4 has no neighbors. The link between nodes 2 and 3 means that they form a FH pair. Since we allow each node to be a member of at most one FH pair, node 1 has no “free” FH neighbors to form a pair with. Likewise, node 4 has no FH neighbors at all and so remains “unpaired” too. From this simple example we can see that the event that some FH node  $i$  forms a pair with its FH neighboring node  $j$  is *not* independent of the status of the other FH nodes from the  $i$  and  $j$ ’s neighborhood. This fact makes the analytical analysis of the FH pairs based solution far more difficult. We will now show how to effectively overcome this difficulty.

<sup>4</sup>An alternative would be to use a similar approach as in the probabilistic key pre-distribution schemes [35], where the nodes would be pre-loaded with a certain number of FH sequences chosen randomly from a common pool.

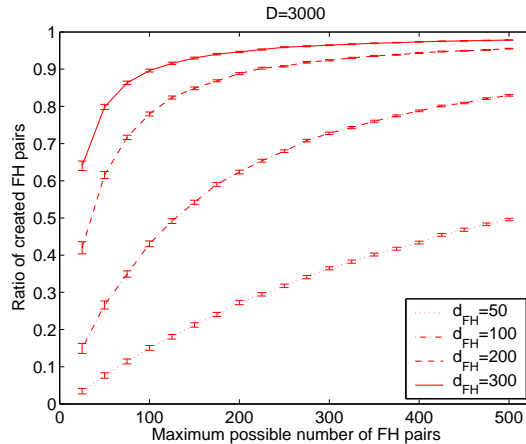


Figure 2.9: Ratio of created FH pairs vs. maximum possible number of FH pairs; we use 95% confidence interval.

## 2.5.2 Analysis of the FH Pairs Based Solution

Again, our goal is to estimate  $P[\text{at least one wormhole} | (x_A, y_A)]$  - the probability that at least one FH pair forms a wormhole from the exposure region to the region not affected by jamming. As we discussed in the previous section, due to the probabilistic nature of the pairing process, not all deployed FH nodes are guaranteed to be a member of some FH pair. To better understand the extent of this potential difficulty, we have conducted the following simulations. We throw randomly a certain number of FH enabled nodes in a deployment region of size  $D \times D$  with  $D = 3000$ . Then we combine FH nodes randomly into FH pairs, with the restriction that a single FH node can be a member of at most one FH pair and two FH nodes can make a pair only if they are within distance  $d_{FH} = \{50, 100, 200, 300\}$  of each other. For each different transmission range and the number of FH nodes, we generate 100 network instances. For each instance we count the number of FH pairs created. The average number of FH pairs, with 95% confidence intervals, is presented on Figure 2.9.

From this figure we can see that except for modest transmission ranges (e.g,  $d_{FH} = 50$ ), the number of created FH pairs is sufficiently high. As expected, the larger the density of the FH nodes is, the larger the number of created FH pairs is. Therefore, with an appropriately selected radio transmission range of FH nodes, we can ensure that almost all the FH nodes will be effectively used.

From the same set of simulations, we have extracted two additional values, namely the average distance between two FH nodes that make a FH pair (the normalized average distance of a FH link) and the corresponding standard deviation. On Figure 2.10, we show the normalized average distance between two FH peers and the corresponding standard deviation as functions of the number of the deployed FH nodes; we normalize the distance with respect to the corresponding radio transmission range  $d_{FH}$ . A striking result on this figure is that the normalized average distance of a FH link is approximately  $0.66 \approx \frac{2}{3}$ , irrespectively of  $d_{FH}$ . Moreover, the standard deviation is approximately 0.23.

This result reminds of the process of picking a random point  $(x, y)$  from the unit circle centered at point  $(x_0, y_0)$ . Then, we can calculate the expected distance  $E[L]$  between points  $(x, y)$  and

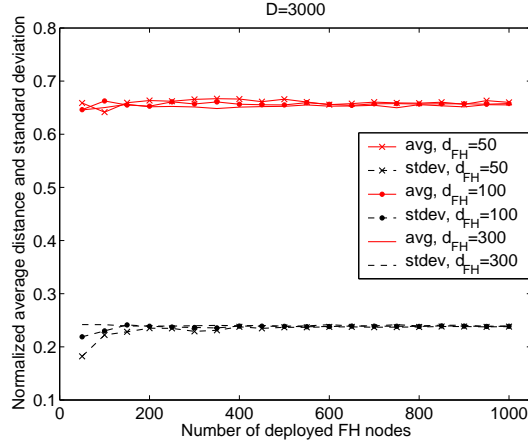


Figure 2.10: Normalized average distance between FH peers vs. the number of FH enabled nodes deployed (“avg” - average, “stdev” - standard deviation).

$(x_0, y_0)$  to be  $E[L] = \frac{2}{3}$  and the standard deviation  $STD(L) = \sqrt{1/18} \approx 0.2357$ . Indeed:

$$\begin{aligned}
 f_L(x) &= \frac{2x\pi}{r^2\pi} = \frac{2x\pi}{1^2\pi} = 2x \\
 E[L] &= \int_0^1 x f_L(x) = \int_0^1 2x^2 = \frac{2}{3} \\
 STD(L) &= \sqrt{\int_0^1 x^2 f_L(x) - (E[L])^2} = \sqrt{\frac{1}{18}}.
 \end{aligned} \tag{2.13}$$

This results suggests that, the random process of opportunistic FH pairing exhibits similar behavior as the process of picking a random point from the circle of radius  $d_{FH}$  centered at the given FH node. To confirm this hypothesis, we have performed another set of experiments. For the given transmission range  $d_{FH}$ , we partition length  $d_{FH}$  into a certain number of mutually exclusive intervals, each of the same size  $\delta$ . Then, we generate a large number of networks (for the fixed parameters  $d_{FH}$ ,  $K$  and  $D$ ) and determine the relative frequency with which distances between created FH pairs fall into each interval. Finally, we compare the relative frequency with the probability of a distance between FH peers falling into the same intervals; we use pdf given in (2.13) to calculate this probability.

As can be seen from Figure 2.11(a) and Figure 2.11(b), the relative frequency matches very well the probability calculated from the postulated probability density function (2.13). This is the case even for low values of  $d_{FH}$  and  $K$ .

This matching inspires the following approach to modelling the creation of a random FH pair in the opportunistic pairing protocol. Consider a FH node  $i$  that is a member of some FH pair. Then, we model the creation of this FH pair, from the FH node  $i$ 's point of view, as picking a random point from the circle with radius  $d_{FH}$ , centered at node  $i$ . Moreover, since FH nodes are deployed randomly and independently of each other, the creation of one FH pair is independent of the creation of another FH pair in the random point picking model. Then, from the independence between different created FH pairs,  $P[\text{at least one wormhole} | (x_A, y_A)]$  can be calculated as follows:

$$\begin{aligned}
 P[\text{at least one wormhole} | (x_A, y_A)] &= 1 - (1 - p_s^{FH})^{K_{FH}} \\
 &\approx 1 - e^{-K_{FH} p_s^{FH}},
 \end{aligned} \tag{2.14}$$

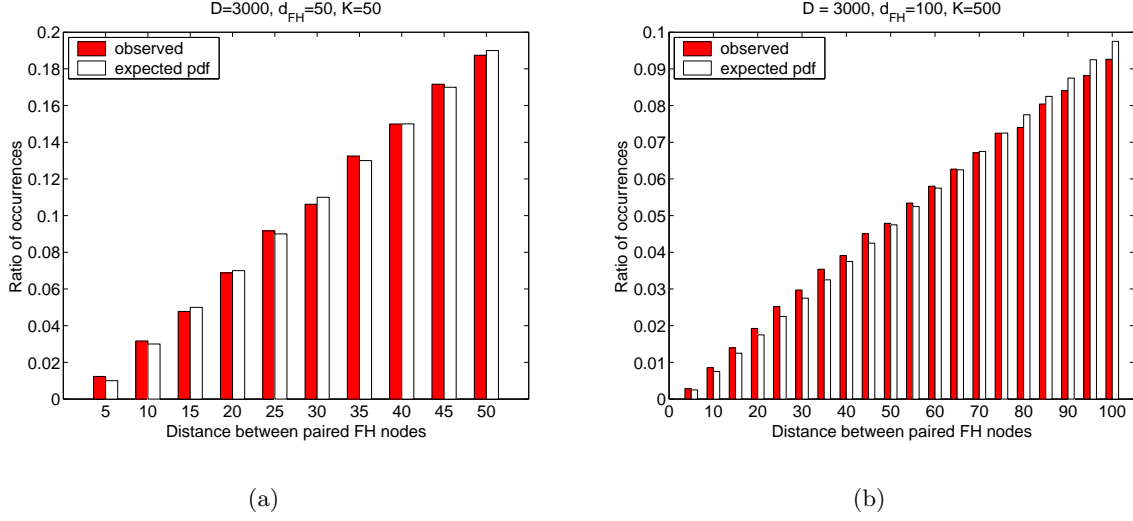


Figure 2.11: Matching between postulated pdf and the relative frequency with which outcomes fall in different intervals of size  $\delta = 5$ : (a)  $d_{FH} = 50$ ,  $K = 50$ , number of experiments=3500; (b)  $d_{FH} = 100$ ,  $K = 500$ , number of experiments=10000.

where  $p_s^{FH}$  is the probability that a single FH pair forms a wormhole and  $K_{FH}$  is the number of created FH pairs.

In order to calculate  $p_s^{FH}$ , we can proceed as in the case of the probability  $p_s$  for wired pairs. However, instead of calculating  $p_s^{FH}$  from scratch, we rather re-use the analytical model developed in Section 2.4.2 by exploiting the similarity between the solution based on wired pairs and the solution based on FH pairs.

Note first that there is a subtle difference in the way we model the deployment of pairs connected through wires and the way we model the creation of FH pairs. In the first case, we use so called “disk line picking” model, i.e., two points are selected randomly and independently from the disk of radius  $\frac{d}{2}$  ( $d$  is the maximum cable length). A well-known result from stochastic geometry says that the expected distance between two randomly selected points from the disk of radius  $\frac{d}{2}$  is  $\frac{128}{45\pi} \frac{d}{2}$  [98]. In the second case, one point (FH node  $i$ ) is given and its FH peer is modelled as a random point selected from the circle of radius  $d_{FH}$ , centered at the location of FH node  $i$ . We have established above that the expected distance between two such selected points is  $\frac{2}{3}d_{FH}$ . Now, the key step in our approximation is that for the given  $d_{FH}$  we scale  $d$  (used in the expressions of Section 2.4.2) in such a way that the expected distances between the random points in the “disk line picking” model and the random points in the model describing the creation of FH pairs are equal, that is,

$$\frac{128}{45\pi} \frac{d}{2} = \frac{2}{3}d_{FH} .$$

From this, it follows:

$$d \approx \frac{d_{FH}}{0.6791} . \quad (2.15)$$

Now, in order to calculate  $P[\text{at least one wormhole} | (x_A, y_A)]$  for the solution based on FH pairs, we first scale  $d$  using expression (2.15) and use  $d$  to calculate  $p_s = P[S]$  (see Section 2.5.3). Then, for the given number of deployed FH nodes, we estimate the average number of created FH pairs

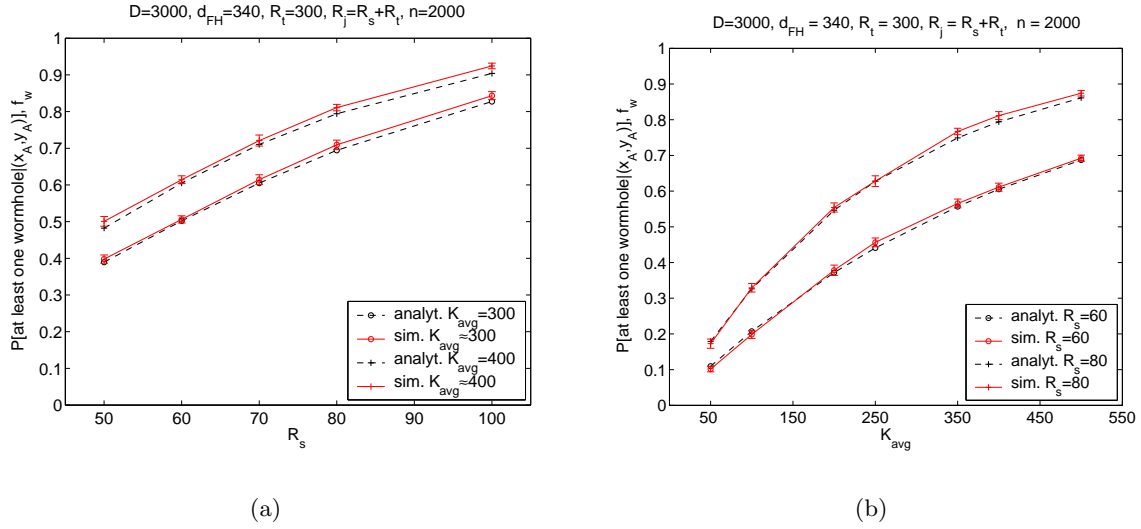


Figure 2.12:  $P[\text{at least one wormhole}|(x_A, y_A)]$  and relative frequency  $f_w(500)$  vs. (a) the size of the exposure region  $R_s$ , and (b) the average number of connected pairs  $K_{avg}$ . We use 95% confidence interval.

(see Figure 2.9) and use this value as  $K$  in expression (2.1). In the following section, we show that this approach works pretty well.

### 2.5.3 Simulations and Model Validation

We investigated the proposed analytical model by means of simulations. We evaluated  $P[\text{at least one wormhole}|(x_A, y_A)]$  as a function of parameters  $K_{FH}$ ,  $R_s$ ,  $d_{FH}$  and  $n$ . As before, we set  $R_j = R_s + R_t$ . For each parameter, we perform 20 experiments as follows. For each different value of a given parameter, we first generate randomly the network topology with  $n$  regular nodes and  $K_{FH}$  FH nodes. To simulate the FH pairing protocol, we iterate randomly through the FH nodes ( $K_{FH}$ ) and for each unmatched FH node  $i$  we try to find another unmatched FH node from  $i$ 's neighborhood. In case node  $i$  has more than one free FH neighbor,  $i$  is matched with a randomly selected one; note that some FH nodes may happen to remain unmatched at the end of the pairing protocol.

Next, we throw randomly  $N = 500$  jamming regions (circles of radius  $R_j$ ) in the deployment area of size  $D \times D$ . Then we count the number  $n_W \leq N$  of jamming regions for which there is at least one wormhole. From this we calculate the relative frequency  $f_w(N) = n_W/N$  for each different value of the given parameter. Finally, we average the results obtained from 20 experiments and present them with 95% confidence interval. To obtain the numerical results, for each value of  $d_{FH}$ , we first scale  $d$  using expression (2.15) and then we plug resulting  $d$  in expression (2.1) to obtain  $P[\text{at least one wormhole}|(x_A, y_A)]$ . The values of  $K$  are obtained as the average number of created FH pairs for different number of FH nodes  $K_{FH}$  (see Figure 2.9).

The results are shown on Figures 2.12-2.13, together with numerical results obtained from the analytical model. In the figures,  $K_{avg}$  represents the average number of created FH pairs. As we can see from the figures, the analytical model predicts quite accurately the probability that at least one wormhole is created. The results obtained have identical properties as in the solution based on



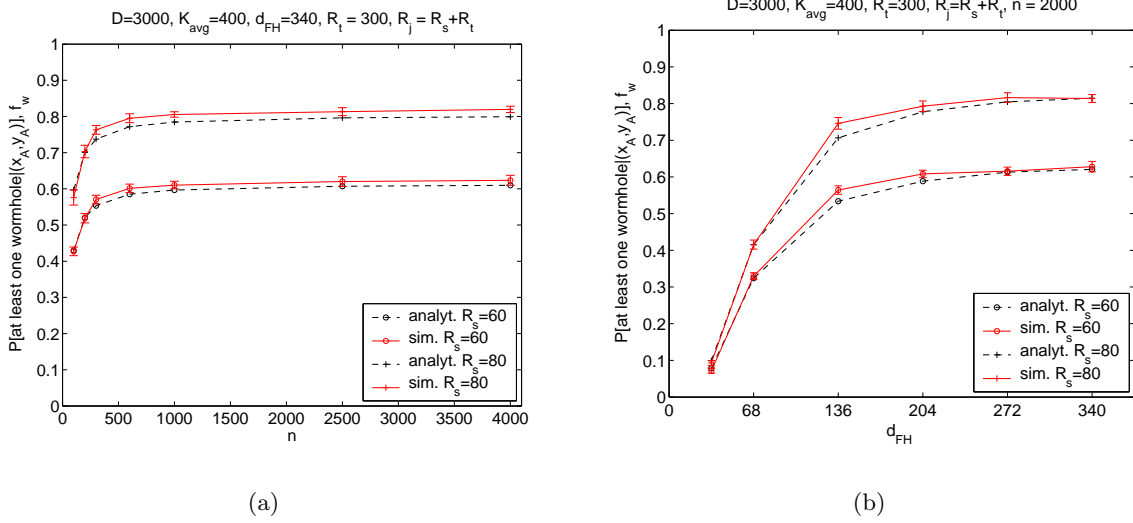


Figure 2.13:  $P[\text{at least one wormhole}|(x_A, y_A)]$  and relative frequency  $f_w(500)$  vs. (a) the number of regular nodes  $n$ , and (b) the transmission range of FH enabled nodes  $d$ . We use 95% confidence interval.

pairs connected through wires. The important difference between wired pairs and FH pairs is that the later achieve the same  $P[\text{at least one wormhole}|(x_A, y_A)]$  with transmission ranges  $d_{FH}$  smaller than the maximum wire length  $d$ ; i.e.,  $d_{FH}/d \approx 0.6791$  (expression (2.15)).

## 2.6 Wormholes via Uncoordinated Channel-Hopping

The solution based on the coordinated FH pairs, though simple, still requires a certain level of synchronization between FH nodes that make a pair. In this section, we explore the feasibility of a completely uncoordinated *channel-hopping* approach. In this solution, we seek to create *probabilistic wormholes* by using sensor nodes that are capable of hopping between radio channels that ideally span a large frequency band. The major difference between channel-hopping (CH) and frequency-hopping is that with the former an entire packet is transmitted on a single channel. In other words, with channel-hopping, sensor nodes hop between different channels (frequencies) in a much slower way (per packet basis), as compared to classical frequency-hopping (e.g., Bluetooth).

### 2.6.1 Rationale of the Approach

In this approach, we can image a part of the deployed nodes or all of them to have channel-hopping capabilities. Regular communication still takes place over a single channel, common to all the nodes. We do not assume channel hopping nodes to be either coordinated or synchronized (see an example of scheduling on Figure 2.14). However, we assume that all the channel-hopping nodes share the common pool of orthogonal channels.

When a channel-hopping sensor node senses the presence of an attacker, it first tries to transmit the report about this event to its neighbors. Each such a report should be acknowledged by intended receivers. In case no (or very few) acknowledgment is received, the node can conclude that an attacker is obstructing his communication. The node then switches to the channel-hopping mode

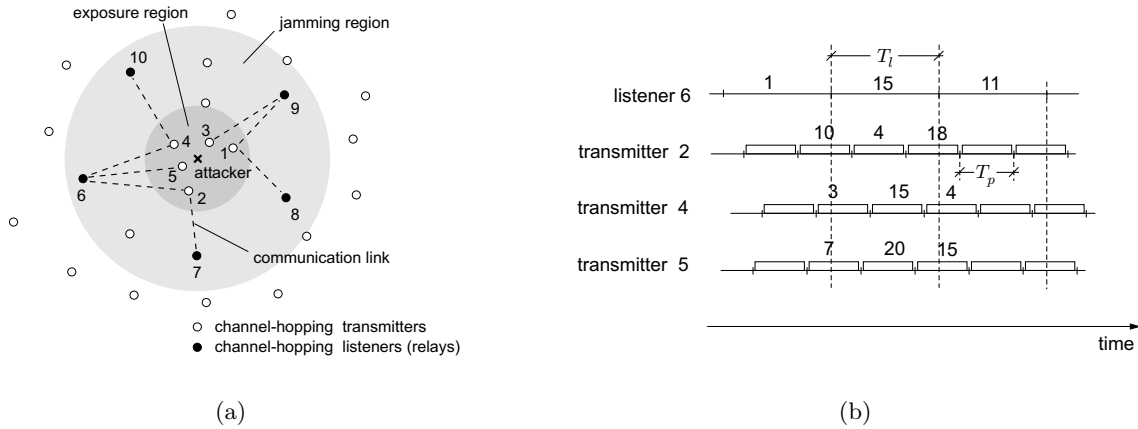


Figure 2.14: (a) A network example with channel-hopping listeners; (b) Example of scheduling for nodes 2, 4, 5 and 6, with  $T_l = 2T_p$  (the numbers above packets represent channel (frequency) indexes).

and repeatedly transmits the same report over different orthogonal channels. In order for this report to potentially be received, the transmitting node has to have at least one neighbor (with channel-hopping capabilities) that listens on one of those channels. Note that we do not assume the two nodes to be synchronized or coordinated. Therefore, the two nodes will happen to occupy the same channel only with some probability; note also that the attacker can potentially jam this channel. Another subtlety of the channel hopping approach is that listening CH nodes enter the channel hopping mode only occasionally (at some predefined rate); we can likewise envision a scenario in which a set of specialized *relaying-only* nodes are deployed. Relaying-only nodes would spend most of the time in the listening mode, hopping randomly between the available orthogonal channels.

When such a node happens to receive the report from the exposure region, it can forward the report further either over the regular channel or by entering in the channel hopping mode.

For this approach to work, we have to ensure that it is not sufficient for the attacker to destroy a whole packet by simply flipping a one or a few bits of the packet. Otherwise, a fast-hopping attacker could easily destroy all the packets transmitted by quickly hopping between the operational channels and jamming every channel for a very short period of time. By encoding packets using appropriate error-correcting codes (e.g., *low-density parity-check* (LDPC) codes), we can achieve a certain level of resistance against jamming [82], which we capture by the notion of a *jamming ratio* (defined in the following section). In this way, we can “keep” the attacker “busy” on one channel for some minimum amount time (that will depend on the jamming radio), while giving an opportunity to transmissions on the other channels to successfully finish.

The implementation of channel-hopping strategies is easily achieved with sensor nodes that use highly programmable software radios (e.g., MICA motes [2]).

## 2.6.2 System Model and Assumptions

Let us first introduce some notation. Let  $I$  denote the set of nodes from the exposure region, which have the channel-hopping capability and which have at least one channel-hopping listening neighbor outside of the exposure region; on Figure 2.14(a),  $I = \{1, 2, 3, 4, 5\}$ . Let  $O$  be the set of channel-hopping listeners that reside outside of the exposure region and that have at least one

channel-hopping transmitting neighbor in the exposure region; on Figure 2.14(a),  $O = \{6, 7, 8, 9, 10\}$ . Also, let  $I_i$  be the set of channel-hopping neighbors from  $I$  of node  $i \in O$ ; on Figure 2.14(a),  $I_6 = \{2, 4, 5\}$ ,  $I_7 = \{2\}$ ,  $I_8 = \{1\}$ ,  $I_9 = \{1, 3\}$  and  $I_{10} = \{4\}$ . We would like to stress here that we are only interested in estimating the chances of a single message being received by any node  $i \in O$ ; the question of whether this is enough to guarantee that this message will be received by the sink we leave for future work.

We assume that there are  $(m+1)$  orthogonal channels available to the sensor nodes. One channel is reserved for the normal mode of operation, i.e., when there is no attack.

We further assume that the nodes from the set  $I$  always transmit (once they sensed the presence of an intruder and have been affected by his jamming activity), while the nodes from the set  $O$  are always in the listening mode (except during a short time period when they have a packet to forward). Both the transmitting nodes and the listening nodes randomly hop between different channels, i.e., the probability of selecting any given channel for the next hop is  $1/m$ . We assume that an attacker knows this strategy, including the channels allocated for hopping.

Further, we denote with  $T_p$  and  $T_l$  the duration of a packet transmitted by node  $i \in I$  and the period during which node  $j \in O$  is listening on a randomly selected channel before switching to another channel, respectively. By setting  $T_l \geq 2T_p$ , we can ensure that even if  $j \in O$  and  $i \in I_j$  are not synchronized, at least one packet of  $i$  will fall within period  $T_l$  of listener  $j$  (see Figure 2.14(b)). In our analysis we set  $T_l = 2T_p$ . We also assume the network nodes to be de-synchronized; indeed, the probability that two nodes are in synch with each other is negligible. The important implication of this assumption is that only one packet of any node  $i \in I$  will fall completely within the given listening period of any listener  $j \in O$ .

We characterize the strength of the attacker by the following three metrics: (i) *channel sensing time*  $T_s$  (i.e. the time the attacker needs to sense if there is a transmission on a channel), (ii) the number of channels  $m_s$  that the attacker can sense simultaneously, and (iii) the number of channels  $m_j$  that the attacker can jam simultaneously. Note that  $m_j$  is limited by the power (energy) that the attacker can afford; in general,  $m_j \neq m_s$ , but in most of our analysis we will assume  $m_j = m_s$ . Clearly, the attacker strength increases with the reduction of  $T_s$ , and the increase of  $m_s$  and  $m_j$ .

We denote with  $T_j$  the minimum jamming period during which the attacker has to jam a given transmission in order to destroy the corresponding packet. We further define the *jamming ratio* ( $\rho_j$ ) as follows,

$$\rho_j \stackrel{def}{=} \frac{T_j}{T_p} \leq 1. \quad (2.16)$$

The higher  $\rho_j$  is, the more resistant are the packets to jamming. Note that our game makes sense only if the jamming ratio is sufficiently high. In [82], Noubir and Lin present a set of different coding strategies (based on *low-density parity-check* (LDPC) codes) that can achieve  $\rho_j = 10 - 15\%$ .

In the sequel, we first study the performance of the proposed system in the case of an inactive attacker (i.e., he does not jam communication). Then, we extend our analysis to scenarios with the active attacker.

### 2.6.3 Performance with an Inactive Attacker

We are interested in calculating the probability  $p_{suc}$  that at least one report about the attacker's presence leaves the exposure region, i.e., is received by at least one listening node  $i \in O$ , after each listener has been listening for the time period of  $T_l$ . We would like to stress again that we are only interested in estimating the chances of a single report being received by any node  $i \in O$ ; the question of whether this is enough to guarantee that the given report will be received by the sink

we leave for future work. Let  $S_i$  denote the event that listening node  $i \in O$  successfully receives a report from any  $j \in I_i$  during one listening period  $T_l$ . Then, we have

$$p_{suc} = 1 - P[\wedge_{i \in O} \overline{S}_i] . \quad (2.17)$$

Due to complex interdependency between different events  $S_i$ ,  $i \in O$ , (i.e., the sets  $I_i$ ,  $i \in O$ , may not be disjoint) calculating this probability leads to an intractable problem. For this reason, we focus on calculating  $p_{suc,i} \stackrel{def}{=} P[S_i]$ , for one fixed listening node  $i \in O$ . Clearly,

$$p_{suc} \geq \max_{i \in O} p_{suc,i} . \quad (2.18)$$

Moreover, if there exists a subset  $O' \subseteq O$  such that  $I_i \cap I_j = \emptyset$ ,  $\forall i, j \in O'$  with  $i \neq j$ , then we have

$$p_{suc} \geq 1 - \prod_{i \in O'} (1 - p_{suc,i}) . \quad (2.19)$$

We have the following result for  $p_{suc,i}$ .

**Proposition 1** *Assuming  $T_l = 2T_p$ , the following holds for any node  $i \in O$*

$$p_{suc,i} = \frac{|I_i|}{m} \left(1 - \frac{1}{m}\right)^{2(|I_i|-1)} . \quad (2.20)$$

*Proof:* Let us denote with  $\mathcal{M}$  the set of available hopping channels (frequencies); note that  $m = |\mathcal{M}|$ . Let a random variable  $F_i \in \mathcal{M}$  represent a channel (frequency) selected by node  $i$ . As discussed above, since  $T_l = 2T_p$ , only one packet of each transmitter  $j \in I_i$  will be contained completely within the observed listening period  $T_l$  of node  $i$  (see Figure 2.14(b)). We call such packets “eligible” packets. Finally, we denote with  $C$  the event that there is a collision between “eligible” packets of at least two transmitters from  $I_i$  on the listening channel  $F_i$ ; in case  $|I_i| = 1$ , the proposition follows trivially.

By the law of total probability we have

$$p_{suc,i} \stackrel{def}{=} P[S_i] = \sum_{k \in \mathcal{M}} P[S_i|F_i = k] P[F_i = k] . \quad (2.21)$$

By assumption,  $P[F_i = k] = m^{-1}$ ,  $\forall k \in \mathcal{M}$ . From this and the fact that transmitters also choose randomly channels to transmit, we have

$$P[S_i|F_i = k] = P[S_i|F_i = k'] , \quad \forall k, k' \in \mathcal{M} . \quad (2.22)$$

Therefore,

$$p_{suc,i} = P[S_i|F_i = k] , \quad \text{where } k \in \mathcal{M} . \quad (2.23)$$

Then, we can write

$$\begin{aligned} P[S_i|F_i = k] &= P[S_i|F_i = k, C] P[C] + P[S_i|F_i = k, \overline{C}] P[\overline{C}] \\ &= P[S_i|F_i = k, \overline{C}] P[\overline{C}] . \end{aligned} \quad (2.24)$$

If there is at least one collision on channel  $F_i = k$ , then we are sure that no “eligible” packet can be received successfully (this is the consequence of setting  $T_l = 2T_p$ ).

The probability  $P[\overline{C}]$  that event  $C$  does not happen is calculated as follows

$$\begin{aligned} P[\overline{C}] &= \sum_{t=0}^1 \binom{|I_i|}{t} \frac{1}{m^t} \left(1 - \frac{1}{m}\right)^{|I_i|-t} \\ &= \frac{|I_i| + m - 1}{m} \left(1 - \frac{1}{m}\right)^{|I_i|-1}. \end{aligned} \quad (2.25)$$

$P[\overline{C}]$  is the probability that either no or only one transmitter from set  $I_i$  transmits on channel  $F_i = k$ .

It remains to calculate  $P[S_i|F_i = k, \overline{C}]$ . Event  $S_i$ , given that listener  $i$  listens on channel  $F_i = k$  and that there are no collisions between “eligible” packets on channel  $F_i = k$ , happens only if there is one “eligible” packet transmitted on channel  $F_i = k$ . For fixed channel  $F_i = k$ , there are

$$K_{suc} = |I_i| (m - 1)^{|I_i|-1} \quad (2.26)$$

different channel allocations such that there is only one “eligible” packet transmitted on channel  $F_i = k$ . The total number of different channel allocations, conditioned on  $F_i = k$  and  $\overline{C}$ , is then

$$K_{tot} = |I_i| (m - 1)^{|I_i|-1} + (m - 1)^{|I_i|}. \quad (2.27)$$

If there were no other packets transmitted by the transmitters from  $I_i$  except the “eligible” packets, then  $P[S_i|F_i = k, \overline{C}]$  would simply be

$$\frac{K_{suc}}{K_{tot}} = \frac{|I_i|}{|I_i| + m - 1}. \quad (2.28)$$

This is because each different channel allocation has the same probability of occurrence (i.e., each node selects a given channel with equal probability  $1/m$ ). Now, since the transmitters from  $I_i$  transmit packets continuously, each “eligible” packet (transmitted by  $j \in I_i$ ) may also experience a collision with at most one “non-eligible” packet per remaining transmitter  $t \in I_i \setminus \{j\}$  (see Figure 2.14(b)). Therefore, we have

$$\begin{aligned} P[S_i|F_i = k, \overline{C}] &= \frac{K_{suc}}{K_{tot}} \left(1 - \frac{1}{m}\right)^{|I_i|-1} \\ &= \frac{|I_i|}{|I_i| + m - 1} \left(1 - \frac{1}{m}\right)^{|I_i|-1}. \end{aligned} \quad (2.29)$$

Finally, the proposition follows by plugging the expressions (2.25) and (2.29) into the expression (2.24).  $\square$

Note that the result in Proposition 1 is conservative, since we assume that any overlapping between two packet transmissions results in a destroyed packet; in general, however, a packet will be destroyed if more than  $\rho_j$  percents of it is affected by a collision (see Section 2.6.2).

We further note that  $p_{suc,i}$  will be equal for every listening period  $T_i$  (this can be seen from Figure 2.14(b)). Thus, the number  $N_{suc,i}$  of listening slots before node  $i$  receives the first report has geometric distribution, that is,

$$P[N_{suc,i} = t] = p_{suc,i} (1 - p_{suc,i})^{t-1}, \quad (2.30)$$

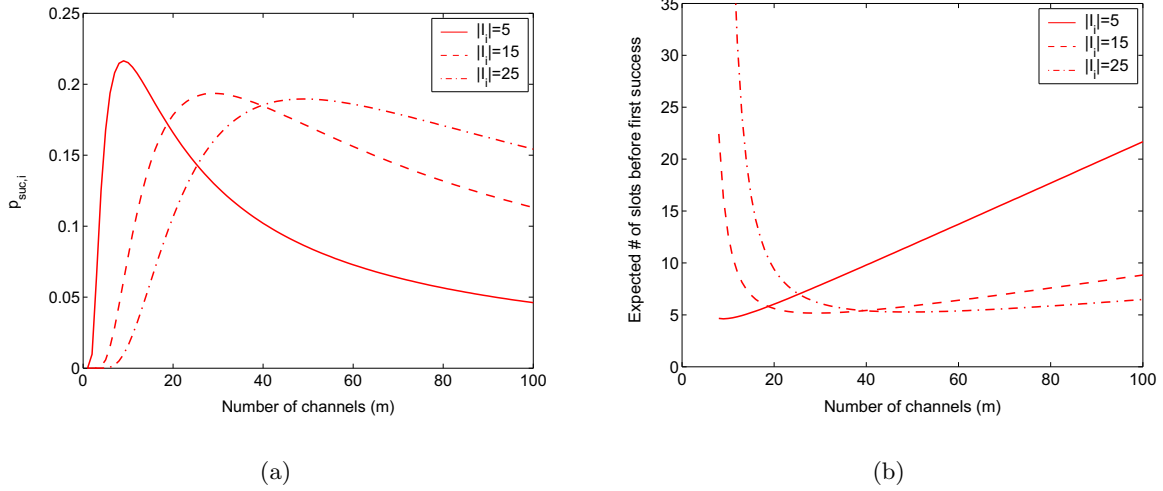


Figure 2.15: Performance with an inactive attacker: (a)  $p_{suc,i}$  vs. the number of channels  $m$ , and (b) the corresponding expected number ( $\bar{N}_{suc,i}$ ) of slots before first success.

and the expected number  $\bar{N}_{suc,i}$  of listening slots before node  $i$  receives the first packet successfully is given by

$$\bar{N}_{suc,i} = \frac{1}{p_{suc,i}}. \quad (2.31)$$

Similarly, the number  $N_{suc}$  of listening slots before any node  $i \in O$  receives the first report successfully also has geometric distribution. Since  $p_{suc} \geq \max_{i \in O} p_{suc,i}$ , we must have

$$\bar{N}_{suc} \leq \bar{N}_{suc,i} \quad \text{and} \quad VAR(N_{suc}) \leq VAR(N_{suc,i}), \quad (2.32)$$

where  $VAR(\cdot)$  denotes variance.

On Figure 2.15(a) and Figure 2.15(b), we plot  $p_{suc,i}$  and the corresponding  $\bar{N}_{suc,i}$ , respectively, for  $|I_i| \in \{5, 15, 25\}$ . Not surprisingly, from Figure 2.15(a) we can observe that there exists an optimal value of  $m$  for fixed  $|I_i|$ . We denote this optimal value with  $m^*$ . Also, the maximum  $p_{suc,i}$  decreases with  $|I_i|$ . It is interesting to observe, however, that the probability  $p_{suc,i}$  increases with  $|I_i|$  for certain values of  $m$  above the optimal point  $m^*$ . Furthermore, on Figure 2.15(b) we can see that  $\bar{N}_{suc,i}$  increases linearly with  $m$  after the optimal point  $m^*$ . Another important observation is that the values of  $\bar{N}_{suc,i}$  are reasonably small. For example, for  $m = 40$  and  $|I_i| = 15$  we have  $\bar{N}_{suc,i} \approx 6$ , meaning that it will take around  $(6 \times T_l) = (12 \times T_p)$  time until the given node  $i \in O$  receives successfully the first report; for as high  $T_p$  as 100 ms, this amounts to only 1.2 seconds (a rather affordable delay for our purposes).

#### 2.6.4 Performance with an Active Attacker

In this section, we are interested in the same performance metrics as in the previous section, with the difference that now the attacker is active (he jams communication). Namely, we want to calculate the probability  $p_{suc,i}^A$  that the given listener  $i \in O$  receives a report within the time period  $T_l$  in the presence of an active attacker. The following holds for  $p_{suc,i}^A$

$$p_{suc,i}^A = p_{suc,i} \times (1 - p_{jam,i}), \quad (2.33)$$

where  $p_{jam,i}$  is the probability that a report (packet) that is potentially successfully received by listener  $i$  is jammed by the attacker (recall that there can be only one such a report per fixed listening period  $T_l$  – see the proof of Proposition 1). The independence in this expression follows from the following facts: (i) from the attacker’s perspective every packet transmitted by the nodes from the exposure region (i.e., set  $I$ ) have equal chance to be received by some listening node  $i \in O$ , and (ii) all the nodes choose their listening and/or transmitting channels independently. It is implicitly assumed in this analysis that the attacker has no information about the set  $I_i$  for the given node  $i \in O$ ; moreover, the attacker has no information about the de-synchronization level between transmitting and listening nodes.

As discussed above, it is sufficient to focus on a single report (packet) that would be successfully received (with probability  $p_{suc,i}$ ) if the attacker was not active. From the moment the transmission of this report commences, the attacker has at most time  $(1 - \rho_j)T_p$  to successfully sense the given report and in turn to jam it.

We assume the following behavior of the attacker. The attacker scans  $m_s$  out of  $m$  channels at a time. If the attacker senses the presence of a signal on a channel, it will jam this channel for the duration of  $T_j$ . The attacker’s success to jam the fixed report (packet) is determined by the “number of attempts”  $k$  that the attacker has available to detect on which channel is the packet being transmitted. To jam successfully the packet, the attacker needs to detect (and jam) the correct channel before a fraction  $(1 - \rho_j)$  of the packet has been transmitted. Therefore, the attacker has at most

$$k = \left\lfloor \frac{(1 - \rho_j)T_p}{T_s} \right\rfloor \quad (2.34)$$

chances to guess on which channel the observed packet has been transmitted. Consequently, the probability  $p_{jam,i}$  that the attacker successfully jams the fixed packet (report) can be upper-bounded as follows

$$p_{jam,i} \leq \frac{k \times m_s}{m}, \quad (2.35)$$

where we assumed  $m_s = m_j$  (Section 2.6.2). This expression is based on the observation that, per duration of  $T_s$ , the attacker can make  $m_s$  guesses about the channel on which the packet is being transmitted. Per packet, this results in  $(k \times m_s)$  guesses. The described behavior of the attacker is the best strategy that the attacker can choose in order to maximize his chances to jam the fixed report. This is because all other strategies reduce the number of guesses  $k$  that the attacker can make to detect the channel on which the observed packet has been transmitted.

We next calculate a lower bound on  $p_{jam,i}$ . For this it suffices to observe that the attacker spends at most  $T_j + T_s$  time per (different) channel visited. Note that this implies that each different channel scanned by the attacker is occupied and therefore the attacker jams it during time  $T_j$ . Therefore, the probability  $p_{jam,i}$  that the attacker successfully jams the fixed packet (report) can be lower-bounded as follows

$$p_{jam,i} \geq \frac{k' \times m_s}{m}, \quad (2.36)$$

where

$$k' = \left\lfloor \frac{(1 - \rho_j)T_p}{T_j + T_s} \right\rfloor \leq k. \quad (2.37)$$

If  $T_s \geq T_j$  and the attacker does not have fast enough hardware to sense the channel and then switch to transmission to jam it, the attacker can employ a different strategy in which he chooses randomly  $m_s$  channels every time  $T_j$  and jams those channels for a duration  $T_j$ . In this case for  $k'$ ,

we have the following

$$k' = \left\lfloor \frac{(1 - \rho_j)T_p}{T_j} \right\rfloor = \left\lfloor \frac{1}{\rho_j} - 1 \right\rfloor \leq k . \quad (2.38)$$

Now, in the general case, we will have a mixture between the two extreme cases discussed above; i.e., the attacker will spend a part of the available time  $(1 - \rho_j)T_p$  sensing the channels that are not active (at a cost  $T_s$ ) and the other part in jamming active channels (at a cost  $T_j$ ). Therefore, in general, the average number  $\bar{k}$  of chances (opportunities) for the attacker to successfully jam the given report satisfies

$$k' \leq \bar{k} = \frac{(1 - \rho_j)T_p}{\bar{T}} \leq k , \quad (2.39)$$

where  $\bar{T}$  is the expected time that the attacker spends per channel visited; note that  $T_s \leq \bar{T} \leq T_j + T_s$ , that is,  $T_s \leq \bar{T} \leq T_j$  in case the attacker does not sense a channel before jamming it.

It is evident from expressions (2.34)-(2.39) that the higher number of occupied channels is the smaller the probability  $p_{jam,i}$  is. In practice, this can be achieved by increasing the number of transmitting nodes, that is, by increasing  $|I_i|$ ,  $i \in O$ . It is important to observe that in this way we decrease the maximum  $p_{suc,i}$  (see Figure 2.15(a)). However, as can be seen from Figure 2.15(a) and Figure 2.15(b), by an appropriate choice of  $m$ , we can indeed afford a significant increase in the number of transmitters while not reducing significantly  $p_{suc,i}$ .

Note finally that all the results in this section are conservative in the sense that we have observed only the performance of a single listening node. Even better performances are expected in the general case that we study in the following section.

### 2.6.5 Simulations

We carried out an evaluation of the solution based on uncoordinated channel hopping using simulations written in Matlab. For the fixed attacker, we are interested in calculating the average number  $\bar{N}_{suc}$  of transmissions until the first report, from the exposure region around the attacker is received by any listening node located outside the exposure region. Here, each time slot is  $T_p$  long, (i.e., equal to the time it takes to a sensor node to transmit a packet).

In our simulations, we consider an *optimal* attacker who knows in advance which channels are to be active; in this way the attacker avoids the cost of visiting non-active channels. However, in these simulations, we consider the case with  $m_s = m_j = 1$ . We have implemented the following strategy for the optimal attacker: every  $T_j$  period, he picks one channel that has not been visited for the longest time among currently active channels.

We perform the following experiment for 20 randomly generated networks of size  $D \times D$ , with  $D = 2000$ . For every network, we first deploy uniformly at random  $N_r$  listening (relaying) nodes and  $N_t$  channel-hopping transmitting nodes. Then, for every network we pick randomly the location of the attacker. The attacker's location, together with the radius of the exposure region  $R_s$  and the radius of the transmission range  $R_t$ , define sets  $I$  and  $O$ ; the set of transmitting nodes residing in the exposure region and the set of neighboring listening nodes located outside of the exposure region).

For each such a scenario and fixed number  $m$  of hopping channels, we generate 50 random (hopping) schedules for both the transmitting nodes (from set  $I$ ) and the listening nodes (from set  $O$ ). Note that the schedules are also shifted in time (the nodes are not synchronized). For every random schedule, we record the time slot at which the first packet from the exposure region is successfully received by any node from  $O$ .



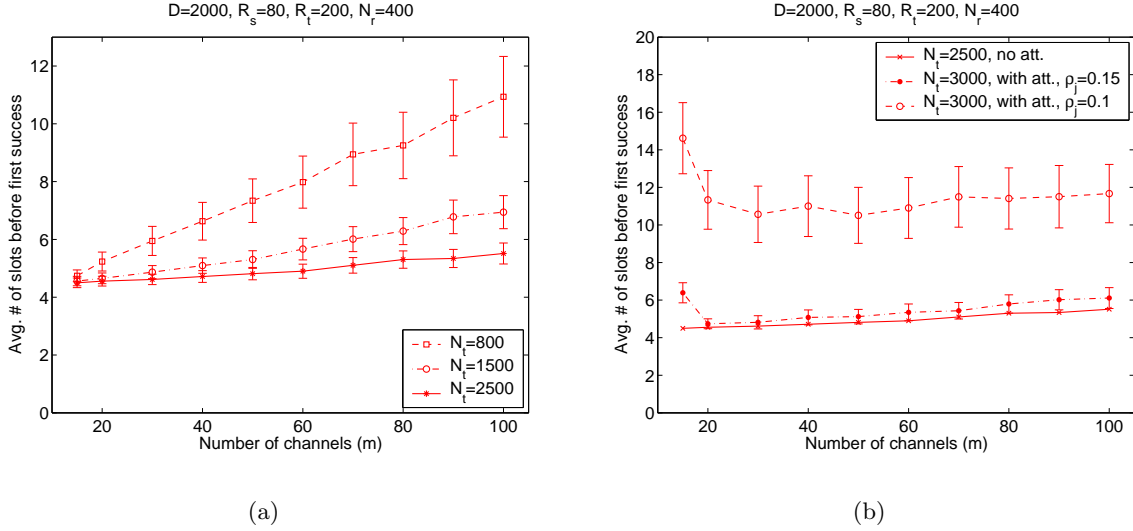


Figure 2.16: Average number  $\overline{N}_{suc}$  of time slots before the first packet is successfully received when (a) the attacker is not active (does not jam), and (b) the attacker is active. We use 95% confidence intervals.

We repeat our experiments for different number  $m$  of hopping channels. For each fixed channel number, we average the results across  $20 \times 50$  experiments described above.

The results are presented on Figure 2.16(a) and Figure 2.16(b), with 95% confidence interval. On Figure 2.16(a), we plot the results for the case when the attacker is not active. From this figure, we can observe that the average number  $\overline{N}_{suc}$  of transmissions before the first success decreases in the number of orthogonal channels. It is important to observe that even if  $m = 1$ , we do not necessarily have collisions at the listening nodes all the time. The reason is that, depending on the node density, for some listening node  $i \in O$ , we will have  $|I_i| = 1$ , with a high probability. Another important observation is that  $\overline{N}_{suc}$  decreases in the density of transmitting nodes from set  $I$  (i.e., in  $N_t$ , for fixed  $D$ ). Finally, the value of  $\overline{N}_{suc}$  is reasonably small, so that we can speak of *timely data delivery* in the approach based on uncoordinated channel-hopping approach. For example, with the communication speed of 19.2 Kbps, the packet size of 20 bytes (including the preamble) and with negligible inter-packet delay,  $\overline{N}_{suc} = 10$  corresponds to approximately 85 ms.

Next we observe  $\overline{N}_{suc}$  in scenarios with an active attacker. The results for jamming ratio  $\rho_j = \{0.1, 0.15\}$  are shown on Figure 2.16(b). Note that  $\rho_j = 0.1$  means that the attacker can jam successfully at most  $1/0.1 = 10$  and  $1/0.15 \approx 7$  packets during time period  $T_p$ . In this figure, the curve obtained for  $N_t = 2500$  and no attacker serves as a reference point. As expected, for the values of  $m$  very close to (or lower than)  $\rho_j^{-1}$ ,  $\overline{N}_{suc}$  grows sharply, essentially meaning that the network will fail to deliver alarms. However, as  $m$  grows above  $\rho_j^{-1}$ , the value of  $\overline{N}_{suc}$  stabilizes at reasonably small value. For example, for  $N_t = 3000$  and  $\rho_j = 0.1$ ,  $\overline{N}_{suc}|_{m=15} = 15$  and  $\overline{N}_{suc}|_{m \geq 20} \approx 11$ . From this figure, we further observe that as we increase the resistance of packets  $\rho_j$  to jamming, we can achieve a significant reduction in  $\overline{N}_{suc}$ .

We also observe that the simulation results are in accordance to our conclusions from Section 2.6.3 and the results shown on Figure 2.15(b). It is interesting to observe in Figure 2.16(a) and Figure 2.16(b) that the variance decreases in both the number of transmitters and  $\rho_j$ , and increases

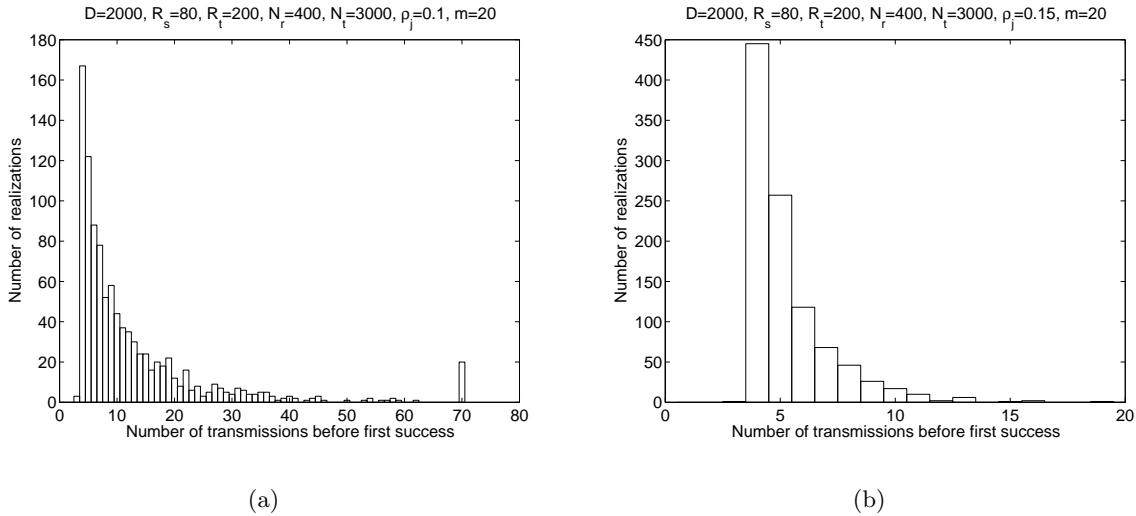


Figure 2.17: Distribution of the “number of transmissions before the first success” for  $m = 20$  and: (a)  $\rho_j = 0.1$ , (b)  $\rho_j = 0.15$ . The number of samples is 1000.

in  $m$ . This is because  $N_{suc}$  is a geometric random variable with the variance given by

$$VAR(N_{suc}) = \frac{1 - p_{suc}}{p_{suc}^2}, \quad (2.40)$$

and  $p_{suc}$  increases in both the number of transmitting nodes for  $m$  above some “critical value” (see Figure 2.15(a)) and  $\rho_j$ , and decreases in  $m$  (Proposition 1 and Figure 2.15(a)).

Another important observation is that the uncoordinated channel-hopping approach is feasible for modest values of  $\rho_j$  and  $m$ , which directly impacts the cost of implementing this approach. This is better seen on Figure 2.17(a) and Figure 2.17(b), where we plot histogram (distribution) of the number of transmissions before the first success ( $N_{suc}$ ) for  $m = 20$ . On Figure 2.17(a), we can observe the jump at  $N_{suc} = 70$ . This is because we round all the realizations with  $N_{suc} > 70$  down to value of 70. These figures further confirm our observation that variance of the  $N_{suc}$  is much higher in the case  $\rho_j = 0.1$  compared to  $\rho_j = 0.15$ ; this can also be seen on Figure 2.16(b). Finally, we can see that the frequency of  $N_{suc}$  indeed resembles geometric distribution.

In conclusion, the approach based on uncoordinated channel-hopping is particularly well suited for timely reporting under jamming attacks. Moreover, being based on uncoordinated channel-hopping, this approach imposes no additional overhead on tiny sensor nodes; it is *zero-configurable*.

## 2.7 Related Work

The issues of jamming detection and prevention in wireless sensor networks have received a significant attention recently. In [34], Wood and Stankovic briefly study potential techniques to avoid jammed regions. A more elaborate study was presented by Wood, Stankovic and Son in [110]. In this work, they propose a proactive protocol that first detects and then maps jammed area. In their approach, each node is assumed to have a detection-module that periodically returns a JAMMED or UNJAMMED message. The message output by the detection module is then broadcast locally. In our approach, we, however, propose reactive solutions that do not require periodic exchange of

information. Xu et. al. [112] propose two countermeasures for coping with jamming: coordinated channel-hopping and spatial retreats, both of which require the nodes to be well synchronized and coordinated. It is not clear that the solution based on spatial retreats is appropriate for wireless sensor networks. In [112], Xu et. al. study the feasibility of reliably detecting jamming attacks. They showed that reliable detection can be a quite challenging task in wireless sensor networks. Moreover, all the proposed detection mechanisms are by their very nature proactive. In [82], Noubir and Lin show how to use low density parity check (LDPC) codes to cope with jamming. In [56], Karlof and Wagner introduce a new attack against wireless sensor networks called sinkholes. It would be interesting to see if sinkholes can be used as a defense mechanism.

There has been a lot of work on DoS attacks in the context of wireless LANs and general ad hoc networks. Bellardo and Savage [15] provide a description of vulnerabilities to DoS attacks in 802.11 management and MAC services; Gupta, Krishnamurthy and Faloutsos [45] study congestion-based DoS attacks. Detection of rational DoS attacks and different countermeasures are proposed by Kyasanur and Vaidya in [63], and more recently by Raya, Hubaux and Aad [90].

## 2.8 Summary

In this chapter, we have made several contributions: we have described in detail how an attacker can mask some events by stealthily jamming an appropriate subset of the nodes. We have further shown how these attacks can be thwarted by means of probabilistic wormholes – the on-demand (reactive) mechanism ensuring timely delivery of important information. We have developed appropriate mathematical models for the presented solutions. Furthermore, we have quantified the probability of success in all the three “probabilistic wormholes”-based approaches.

It is clear that the space of investigation of this area is huge: other approaches can be envisioned, and for the three that we have presented, the influence of other parameters can be studied. Yet, we believe that this work provides useful insights on how to quantify the effectiveness of wormhole-based defense mechanisms.

In terms of future work, it would be interesting to evaluate the performance of hybrid solutions, by combining the three approaches proposed in this chapter. Finally, it would be interesting to implement the presented schemes.



## Chapter 3

# Key Agreement in Peer-to-Peer Wireless Networks

### 3.1 Introduction

As the popularity of mobile systems such as PDAs, laptops, and mobile phones increases daily, users tend to rely on them in a growing number of situations. In this chapter, we focus on the frequent case in which two people get together (e.g., at a meeting, or in the street) and make use of their devices to communicate with each other, or at least to exchange their (electronic) business cards. Clearly, the communication between these devices must be properly secured.

Very often, the two users will want the security between their devices to be peer-to-peer, thus operating independently from any authority. In practice, this means that the mobile devices must run a protocol to authenticate each other and to protect the data they exchange (to ensure confidentiality and integrity); the latter operation typically requires setting up a symmetric shared key. This key can be used to secure both immediate communications and communications that take place afterwards (e.g., when users exchange e-mail over the Internet).

It is a common belief that peer-to-peer security is more difficult to achieve than traditional security based on a central authority; moreover, wireless communication and mobility are considered to be at odds with security. Indeed, jamming or eavesdropping is easier on a wireless link than on a wired one, notably because such mischief can be perpetrated without physical access or contact; likewise, a mobile device is more vulnerable to impersonation and to denial-of-service attacks.

In contrast to this widespread belief, we think that physical presence is the best way to increase mutual trust and to exchange information in a secure way. Indeed, authentication is straightforward, as users can visually recognize each other (if they meet for the first time, they can be introduced to each other by a common friend whom they trust; or they can check each other's ID). In order to establish a shared key, they can make use of a location limited channel (e.g., physical contact or infrared [101, 14]) between their two devices. The man-in-the-middle attack is considered to be infeasible in these conditions.

Recently, researchers have proposed solutions that run exclusively on a radio link (hence they do not require a special channel such as physical contact or infrared), which increases usability. To compensate for the much higher vulnerability of radio channels, in some solutions users are required to type a password in both devices [43]; in other solutions, they simply have to compare strings of words (the longer the string, the higher the security) [43, 50, 24].

In this chapter, we describe two novel approaches to the problem of *user-friendly* key agreement

(and mutual authentication) in settings where the users do not share any authenticated secret or certified public key in advance. The first approach belongs to the family of solutions requiring the users to compare strings of words, whereas the one is completely novel; it is based on radio-channel specific technique called *distance-bounding*. In addition, we make the following contributions: (i) we design protocols that are provably secure in a realistic communication model, (ii) we apply a modular approach to designing and analyzing the protocols, thus paving the way to the design of *re-usable* (provably secure) message transfer (MT) authenticators, and (iii) we significantly increase user-friendliness. In the same vein, in Chapter 4, we develop *Integrity (I)-codes* that provide a method to ensure the integrity (and authentication) of a message transmitted over insecure channel, and which are based solely on message coding (no shared secret or certified public key is required).

The chapter is organized as follows. In Section 3.2 we state the problem and the set of assumptions. In Section 3.3, we describe a novel message transfer authenticator. In Section 3.4, we show the application of the developed message transfer authenticator to secure key agreement. In Section 3.5, we provide the security analysis of the message transfer authenticator. In Section 3.6, we describe a key agreement protocol based on verifiable principal proximity (distance bounding). We overview the related work in Section 3.7. Finally, we summarize the chapter in Section 3.8.

## 3.2 Problem Statement and Assumptions

We consider the following problem. Two users, each equipped with a personal device capable of communicating over a radio link, get together and want to establish a shared key. Although they can visually recognize each other, we assume that they do not share any authenticated cryptographic information (e.g., public keys or a shared secret) prior to this meeting. In addition, the users can communicate only over an *insecure* radio channel (no infrared or physical ports are available). The challenge is the following: *How can the users establish a shared key in a secure way?*

### 3.2.1 Threats Against Radio-Based Systems

The Diffie-Hellman (DH) key agreement protocol [104] seems to be appropriate for the problem (and the set of assumptions) at hand; the DH key agreement protocol is believed to be secure against a passive adversary<sup>1</sup> (e.g., eavesdropping on a wireless link). Let us briefly review how the DH key agreement protocol works. To agree on a shared key, two users, Alice ( $A$ ) and Bob ( $B$ ) proceed as follows.  $A$  picks a random secret exponent  $X_A$ , and calculates the DH public parameter  $g^{X_A}$ , where  $g$  is a generator of a group of large order.  $B$  does the same, that is, he calculates  $g^{X_B}$ . Finally,  $A$  and  $B$  exchange the public parameters  $g^{X_A}$  and  $g^{X_B}$  and calculate the shared DH key as  $K = g^{X_A X_B} = (g^{X_A})^{X_B} = (g^{X_B})^{X_A}$ .

It is well known that the basic version of the protocol is vulnerable to an active adversary who uses a *man-in-the-middle* (MITM) attack. At first glance, it may seem that mounting the MITM attack against wireless devices that communicate over a radio link and are located within the radio communication range of each other can be perpetrated only by a sophisticated attacker. But this is not the case, as we will now explain by a simple example in the framework of Internet protocols.

The Address Resolution Protocol (ARP) [87] is used by the Internet Protocol (IP) to map IP network addresses to the hardware addresses used by a data link protocol. An attacker can send spoofed ARP-replies to the victim, who will consequently send all its packets to the attacking machine. In an experiment we conducted, we were able to redirect the traffic between two “legal”

---

<sup>1</sup>This is true if the Computational Diffie-Hellman problem [79] is intractable.

machines through an attacking machine, despite the fact that the two legal machines were in radio-communication range of each other. In this way, the attacker could perpetrate the MITM attack (by altering the DH parameters). For this attack we used a collection of publicly available tools for network auditing and penetration testing, called *dsniff* [100, 99].

Of course, ARP-spoofing is not the only way to mount a MITM attack against wireless devices. Examples of more involved MITM attacks against Bluetooth [7] equipped devices can be found in [52] and [62].

Hence, our goal is to devise mechanisms that prevent the attacker from modifying the DH parameters without being noticed.

### 3.2.2 Assumptions

We assume the users to be equipped with a computationally constrained personal device (e.g., a PDA). Each device is equipped with a radio transceiver (e.g., IEEE 802.11 [6]). We also assume that each device has a human-friendly interface (i.e., a screen and a keyboard). Furthermore, the user are able to exchange a short information (string of bits) over a *low-bandwidth authentication* channel like personal *voice* or *visual* communication (e.g., by looking at each other's displays).

We would like stress that a part of the protocols proposed in this chapter are not limited to the setting where two users meet in person. Our approach is also applicable (easily extensible) to, for example, communication over the Internet, given that the users have a means to realize a low-bandwidth authentication channel (e.g., authentication of the voice over a phone or Short Message Service (SMS)).

We will present our solution over the multiplicative group  $\mathbb{G}$  with the generator  $g$ . Here, we take  $\mathbb{G}$  to be a subgroup of  $\mathbb{Z}_p^*$  of the prime order  $q$ , where  $\mathbb{Z}_p^*$  is the multiplicative group of non-zero integers modulo a large prime  $p$ . However, the whole treatment here applies to any group in which the Decisional Diffie-Hellman (DDH) problem is hard. These are all groups in which it is infeasible to distinguish between quadruples of the form  $(g, g^x, g^y, g^{xy})$  and quadruples  $(g, g^x, g^y, g^z)$  where  $x, y, z$  are random exponents. Furthermore, we assume that  $p$  and a generator  $g$  of  $\mathbb{Z}_p^*$ , ( $2 \leq g \leq p-2$ ) are selected and published. All devices are preloaded with these values<sup>2</sup>.

### Adversarial Model

In this chapter, we assume that the adversary Mallory ( $M$ ) controls the radio communication channel in a sense that he can obtain messages and modify transmitted messages by adding her own messages to the channel. We further assume that the adversary cannot disable the communication channel over an alternative low-bandwidth channel, that is, the adversary cannot prevent the users to exchange information over voice or visual communication. This assumption simply reflects everyday situations where people meet each other and talk.

Our adversarial model is similar to the Dolve-Yao model [79] in that the adversary controls the radio communication channel can initiate a conversation with any other user. But it differs in that we assume that the adversary cannot disable the communication over low-bandwidth authenticated channels (he cannot prevent the users to talk to each other or to look at each other's displays).

We further assume  $M$  to be *computationally bounded*. Also, the two parties involved in the communication do trust each other; otherwise, little can be done (a corrupted party can always

---

<sup>2</sup>We stress here that we could let users select and communicate to each other their own parameters  $p$  and  $g$ . However, this would come at the expense of the number (and size) of messages to be exchanged between the users, and our goal is to keep key exchange protocols as simple as possible.

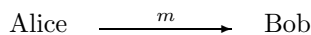


Figure 3.1: Message authenticator in the “ideal world” (with passive adversaries).

disclose any secret information received by another party). Whenever we speak of the security of a given protocol, we implicitly assume that the users involved in the protocol (e.g., their devices) are not compromised.

### 3.2.3 Commitment Schemes

Commitment schemes are an important cryptographic building block that we will be using in our protocols. In this subsection, we provide only an informal treatment of commitment schemes. The semantics of a commitment scheme are the following: (i) a user who commits to a certain value cannot change this value afterwards (we say that the scheme is *binding*), (ii) the commitment is hidden from its receiver until the sender “opens” it (we say that the scheme is *hiding*).

A commitment scheme transforms a value  $m$  into a commitment/opening pair  $(c, d)$ , where  $c$  reveals no information about  $m$ , but  $(c, d)$  together reveal  $m$ , and it is infeasible to find  $\hat{d}$  such that  $(c, \hat{d})$  reveals  $\hat{m} \neq m$ . Now, if Alice wants to commit a value  $m$  to Bob, she first generates the commitment/opening pair  $(c_A, d_A) \leftarrow \text{commit}(m)$ , and sends  $c_A$  to Bob. To open  $m$ , Alice simply sends  $d_A$  (and  $m$  if necessary) to Bob, who runs  $\hat{m} \leftarrow \text{open}(\hat{c}_A, \hat{d}_A)$ ; we denote with  $\hat{x}$  the message at the receiver’s side when message  $x$  is sent over a public (unauthentic) channel. If the employed commitment scheme is “correct”, at the end of the protocol we must have  $m = \hat{m}$ . We are now ready to describe our protocols.

## 3.3 Message Transfer (MT) Authenticator

Let us first define the notion of a *message transfer authenticator*. Let us consider a simple example on Figure 3.1. Here Alice ( $A$ ) sends a message  $m$  to Bob ( $B$ ) over some link. Any protocol (or, more generally, mechanism) that ensures that the message accepted by Bob is the same message that has been sent by Alice, except with a satisfactorily low probability, is called a message authenticator.

### 3.3.1 Straightforward Approaches are Suboptimal or Flawed

Perhaps the simplest way to verify the validity of the received message for Bob is to report the received message to Alice who in compares it against the messages  $m$  she sent previously. The comparison of the message value can be performed by looking at the screen of the communicating party, or by reading aloud the value to be compared. Although this approach provides very strong security, it is clearly impractical because it is limited to very short strings. A possible way to make visual (and verbal) verification easier for  $A$  and  $B$  is to represent the message in a more readable form by, for example, significantly reducing the number of digits to be compared (and potentially encoding the bits in a more readable form as in RFC 2289 [4]). However, in this way, many different (long) DH public parameters translate to the same (short) bit string (the check value). This may give some advantage to a potential attacker.

Another simple approach consists in first exchanging  $m$  over a public channel, and in turn, verifying (for example, visually) that  $h(m)$  matches  $h(\hat{m})$ , where  $h$  is a hash function satisfying certain security properties and  $\hat{m}$  is the message as received by  $B$ . In order for this approach to be usable, the output of the hash function  $h(\cdot)$  should be truncated to a relatively short length (e.g.,



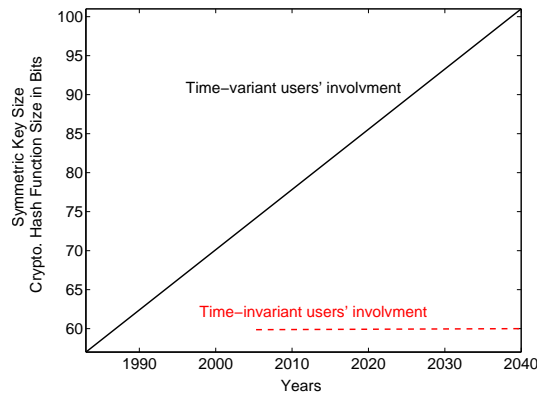


Figure 3.2: Time-variant versus time invariant users’ involvement.

around 50 bits). However, the adversary can easily find a collision on the truncated output of short length. In Section 3.7, we will describe a flaw in an approach proposed by [77], where two users perform DH key exchange, using the public DH keys  $g^{X_A}$  and  $g^{X_B}$ , and at the end of the protocol verify the validity of the shared key by comparing the truncated output of a hash function applied to the shared key  $K = g^{X_A X_B}$ .

Another important aspect of user-friendly authentication mechanisms is that of *time invariance*. For example, the straightforward solution outlined in the previous paragraph has the drawback that the size of the cryptographic hash function  $h(\cdot)$  increases over time, in order to compensate for fast (daily) advance in computational technology and computational power available to an adversary. Today a “target collision-resistant” hash function (used in the previous example) implies the hash function size of at least 80 bits [68] (see Figure 3.2). However, according to [68], the minimum required size increases linearly over time due to fast technological advances (as shown in Figure 3.2). Consequently, straightforward solutions similar to the one outlined above imply *time-variant users’ involvement*, that is, the number of bits to be compared by users increases over time. This is certainly not a desirable property and therefore in this chapter we look for a solution that exhibits the *time-invariant users’ involvement* property.

In order to make the approach based on string comparison usable, it is essential to make a *proper trade-off between security and usability*. In the following section, we propose a provably secure MT-authenticator, called MT-SC (MT-authenticator with String Comparison) that achieves *optimal* trade-off between security and usability. Moreover, the MT-SC is a time-invariant solution; in spite of the fact that the sizes of the used cryptographic primitives (their security parameters) increase with time, the users’ involvement remains unchanged (see Figure 3.2). The MT-SC is a basic building block of the key agreement protocols that we propose in this chapter.

### 3.3.2 Optimal MT-Authenticator Based on String Comparison

The MT-SC authenticator is shown on Figure 3.3. Alice holds a message  $m$  (not necessarily secret) that she wants to send to Bob. Both  $A$  and  $B$  first generate  $k$ -bit random strings  $N_A$  and  $N_B$ , respectively. Then,  $A$  calculates commitment/opening pair  $(c, d)$  for the concatenation

$$m || N_A .$$

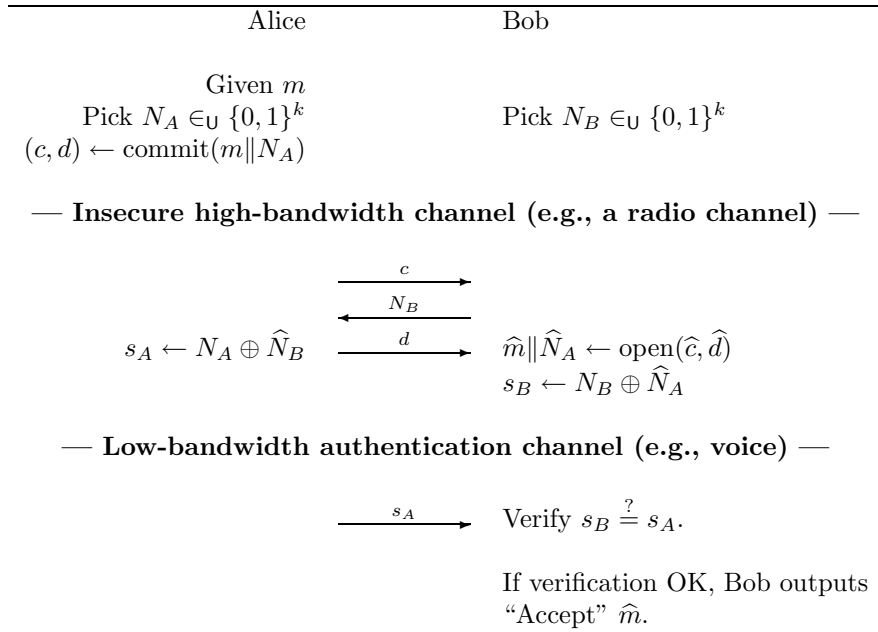


Figure 3.3: Message authenticator based on short strings comparison (MT-SC). Bob checks whether the source of the message  $\widehat{m}$  is Alice; an “authentic channel” can be implemented through visual or vocal comparison of the output strings  $N_A \oplus \widehat{N}_B$  and  $N_B \oplus \widehat{N}_A$ .

The following three messages are exchanged over an insecure radio link. In the first message,  $A$  sends to  $B$  the commitment  $c$ .  $B$  responds with his random string  $N_B$ . In turn,  $A$  sends out  $d$ , by which  $A$  opens the commitment  $c$ .  $B$  checks the correctness of the commitment/opening pair  $(\widehat{c}, \widehat{d})$ . If the verification is successful,  $A$  and  $B$  proceed to the final phase (Figure 3.3).

In the final phase,  $A$  and  $B$  first generate the verification strings  $s_A$  and  $s_B$ , respectively, as shown on Figure 3.3 ( $\oplus$  is the bitwise “xor” operation). The length of each of these strings is  $k$ . Finally, Alice and Bob (as users) simply compare  $s_A$  and  $s_B$  (e.g., through visual or vocal communication). If they match, Bob accepts the message  $\widehat{m}$  as being authentic (i.e., he accepts that  $\widehat{m} = m$ ).

We assess the security of our MT-authenticator in Section 3.5. Here, we only state the result. To do this, we define formally what we mean by a secure MT-authenticator.

We denote with  $\Pi(k, (A, B))$  a message authentication protocol (an MT-authenticator) that is executed between two parties  $A$  and  $B$ , with the security parameter  $k$ . Here, by authentication of a message we mean that at the end of a successful run of the protocol, party  $B$  accepts that a message  $\widehat{m}$  it has received must have been sent by party  $A$ , except with a negligible probability.

**Definition 5** *We say that  $\Pi(k, (A, B))$  is a secure message authentication protocol (MT-authenticator) between  $A$  and  $B$  if adversary  $M$  cannot win, except with a satisfactorily small probability  $O(2^{-k})$ .*

Informally,  $M$  wins if  $B$  reaches the “Accept”  $\widehat{m}$  decision while  $\widehat{m}$  has not been sent by  $A$ .

**Perfectly Hiding and Computational Binding Commitment Scheme.** For simplicity, in this chapter, we will assume that the used commitment scheme  $\text{commit}(\cdot)$  is perfectly hiding and computational binding; we would like to emphasize, however, that the protocol given in Figure 3.3

works with any commitment scheme. We next define more formally basic properties of such commitment schemes. In this direction, let us define the following event  $B$  for the fixed commitment scheme:

$$B \stackrel{\text{def}}{=} \{ \text{any probabilistic polynomial time algorithm (adversary) outputs a commitment } (c) \text{ and two valid distinct openings } (d \text{ and } d', d \neq d') \} .$$

**Definition 6** *We say that the commitment scheme is perfectly hiding and computational binding if (i)  $P[B] \stackrel{\text{def}}{=} \varepsilon(\text{par})$  is negligible in the security parameter(s)  $\text{par}$  of the commitment scheme, and (ii) commitments reveal no information whatsoever about the committed values.*

We next analyze the security of the MT-SC protocol (Figure 3.3) with the  $\text{commit}(\cdot)$  being a perfectly hiding and computational binding commitment scheme. Let us denote with  $\overline{B}$  the complementary event to the event  $B$ . In other words,  $\overline{B}$  says that no probabilistic polynomial time algorithm (adversary) outputs a commitment  $(c)$  and two valid distinct openings  $(d \text{ and } d', d \neq d')$ . Furthermore, let us define the event  $S$  as follows

$$S \stackrel{\text{def}}{=} \{ \text{the adversary wins (succeeds) against the given user of the MT-SC (with the } \text{commit}(\cdot) \text{ being perfectly hiding and computational binding) in polynomial time} \} .$$

Now we can write the following:

$$\begin{aligned} P[S] &= P[S|B]P[B] + P[S|\overline{B}]P[\overline{B}] \\ &\leq P[B] + P[S|\overline{B}] \\ &= \varepsilon(\text{par}) + P[S|\overline{B}] . \end{aligned} \tag{3.1}$$

The expression (3.1) says that we can focus our security analysis on the case where the event  $B$  never happens ( $S|\overline{B}$ ), that is, the adversary does not break the binding property of the commitment scheme. In other words, we can condition the success of the (computationally bounded) adversary on the event  $\overline{B}$  and evaluate the probability  $P[S|\overline{B}]$ , and finally add up  $P[S|\overline{B}]$  to the probability  $\varepsilon(\text{par})$ .

It is interesting to note that since we assume the used commitment scheme to be perfectly hiding, the event  $S|\overline{B}$  essentially means that the semantics (the basic functionality) of commitment schemes is “unconditionally” preserved, as if an *ideal commitment scheme* exists and is used in our protocol.

Let us denote with  $\gamma$  the maximum number of sessions (successive or abortive) of the MT-SC protocol by the given user. It is important to note that the security parameters  $\text{par}$  of the used commitment scheme impose upper bounds on the security parameters  $k$  and  $\gamma$  in our protocol. For example, if for the used commitment scheme the probability  $\varepsilon(\text{par})$  is negligible given that the number of executions of this scheme is bounded by  $q$  (where  $q$  is one of the parameters  $\text{par}$ ), then it does not make sense to consider  $\gamma > q$ .

**Theorem 9** *The probability  $P[S|\overline{B}]$  that an adversary succeeds (wins) against a targeted user of the MT-SC protocol (with the  $\text{commit}(\cdot)$  being perfectly hiding and computational binding) conditioned on the event  $\overline{B}$  is bounded by  $\gamma 2^{-k}$ , i.e.,  $P[S|\overline{B}] \leq \gamma 2^{-k}$ .*

We prove this theorem in Section 3.5. Then, by combining the result from Theorem 9 and the last expression in (3.1) we have the following result.

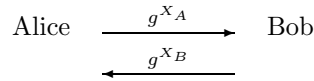


Figure 3.4: Basic Diffie-Hellman key agreement protocol is secure in the AM model; for simplicity, we drop  $(\text{mod } g)$  from the DH public keys  $g^{X_A}$  and  $g^{X_B}$ .

**Theorem 10** *The probability  $P[S]$  that an adversary succeeds (wins) against a targeted user of the MT-SC protocol (with the  $\text{commit}(\cdot)$  being perfectly hiding and computational binding) is bounded by  $\gamma 2^{-k} + \varepsilon(\text{par})$  (i.e.,  $P[S] \leq \gamma 2^{-k} + \varepsilon(\text{par})$ ), where  $\varepsilon(\text{par}) = P[B]$  is negligible in the security parameter(s)  $\text{par}$  of the used commitment scheme. Therefore, for appropriately chosen parameter  $k$ , MT-SC is a secure MT-authenticator.*

Let us give an example of possible values for the parameters  $k$  and  $\gamma$ . Let us assume that the fixed user can participate in at most  $\gamma = 2^{20}$  sessions (successful or abortive) in his/her lifetime; this corresponds to 32 sessions per day during approximately 89 years. Then, by choosing  $k = 50$  (bits) we obtain that the highest probability of success by the adversary (having seen a huge number of  $2^{20}$  MT-SC sessions by the fixed user) is approximately  $\gamma 2^{-k} = 2^{-30}$ . Note that  $k$  also represents the length of the verification strings  $s_A$  and  $s_B$  to be compared by users. To make this job easier for users, we can encode  $k = 50$  bits into a certain number of short words from some predefined dictionary (e.g., RFC 2289 [4]).

The proposed MT-authenticator (Figure 3.3) is *optimal* in the sense that all the  $k$  bits to be compared by the users running this protocol contribute to the uncertainty of the adversary. It is also *time-invariant* in the sense that the users involvement ( $k$ ) does not increase with time. These two are distinguishing features of our optimal MT-authenticator compared to any other proposal (see Section 3.7).

In the following section, we show how to use the optimal MT-authenticator from Figure 3.3 in conjunction with the *basic* Diffie-Hellman key agreement protocol to build a secure key agreement protocol, where the users' involvement brought to merely comparing a  $k$ -bit string.

### 3.4 From Secure MT-Authenticator to Secure Key Agreement

In [16], Bellare, Canetti, and Krawczyk propose a very intuitive modular approach to security analysis (and construction) of secure protocols. This approach assumes two adversarial models: the *authenticated link model (AM)* and the *un-authenticated links model (UM)*. The AM model is an ideal-world model in which the attacker is able to invoke protocol runs, masquerade as protocol principals, and find old session keys; however, he is not able to fabricate or replay messages that appear to come from uncorrupted parties. The UM model is a real-world model, in which the attacker can do all that it can in the AM model; in addition, he can replay messages and try to fabricate messages.

The security of the protocol is first proven in the AM model, assuming (as assumed by the model itself) that all the communication between the parties is authenticated. If the protocol is proven to be secure in the AM model, then it can be shown to be secure in the UM model, provided that each message transmitted between the parties is authenticated by a MT-authenticator. Using this approach, in [16], Bellare, Canetti, and Krawczyk show that the basic Diffie-Hellman protocol is secure in the AM model (Figure 3.4), and that it is secure in the UM model, provided that correct MT-authenticators are used to authenticate transfers of DH public parameters. In their work, they

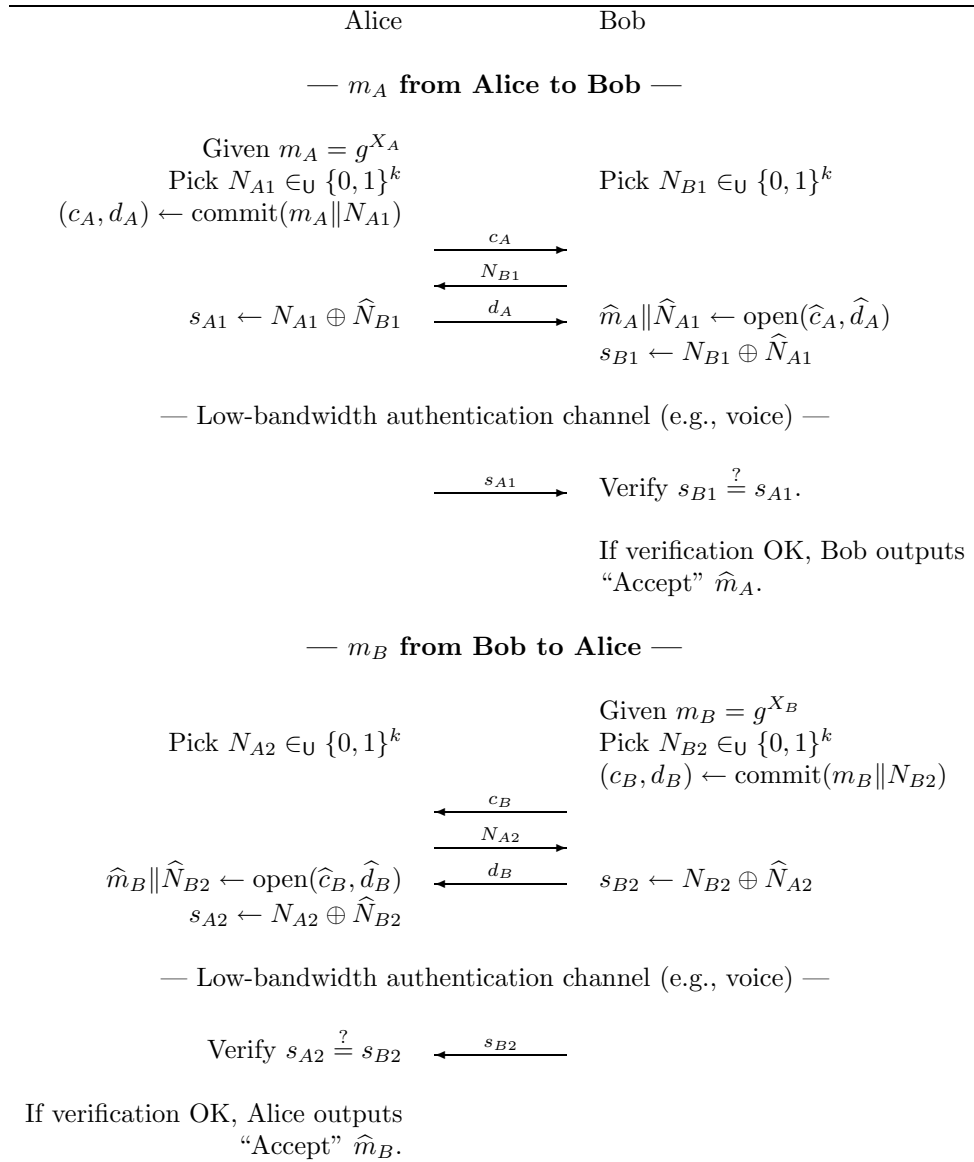


Figure 3.5: Straightforward application of the MT-SC authenticator to basic DH key agreement protocol.

use MT-authenticators based on digital signatures and encryption.

In this chapter, we follow exactly the same approach, but instead relying on digital signature, as a basis for an MT-authenticator, we use the MT-SC authenticator developed in the previous section (cf. Figure 3.3).

### 3.4.1 Straightforward Application of the MT-SC Authenticator

A straightforward application of the MT-SC authenticator to the basic DH key exchange protocol results in a (secure) protocol that involves 6 messages and 2 string comparisons as can be seen on Figure 3.5. The security of the key exchange protocol on Figure 3.5 follows directly from the security of the MT-SC authenticator (Theorem 10); the DH public keys  $g^{X_A}$  and  $g^{X_B}$  are simply

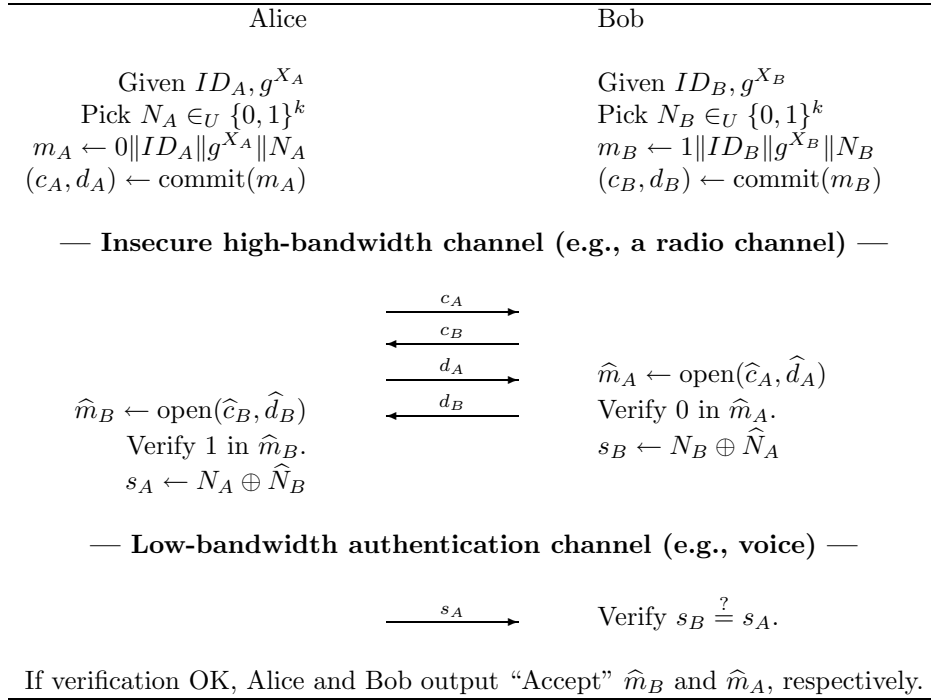


Figure 3.6: Optimal Diffie-Hellman key agreement protocol based on String Comparison (DH-SC)

treated as message  $m$  in the MT-SC protocol from Figure 3.3. While being secure, this protocol is still somewhat suboptimal, as it requires 6 messages to be exchanges over the insecure (radio) channel and, more importantly, it involves two  $k$ -bits comparisons.

In the following section, we present an optimized version of the secure DH key agreement protocol from Figure 3.5, which involves 4 messages over an insecure channel and only one  $k$ -bit string comparison.

### 3.4.2 Diffie-Hellman Key Agreement with String Comparison (DH-SC)

The optimized protocol unfolds as shown on Figure 3.6. Both Alice ( $A$ ) and Bob ( $B$ ) have selected their secret exponents  $X_A$  and  $X_B$ , respectively, randomly from the set  $\{1, 2, \dots, q\}$  ( $q$  being the order of  $\mathbb{G}$ ) and calculated DH public parameters  $g^{X_A}$  and  $g^{X_B}$ , respectively.  $A$  and  $B$  proceed by generating  $k$ -bit random strings  $N_A$  and  $N_B$ , respectively. Finally,  $A$  and  $B$  calculate commitment/opening pairs for the messages  $m_A$  and  $m_B$ , respectively, where

$$\begin{aligned}
 m_A &\leftarrow 0 \| ID_A \| g^{X_A} \| N_A \\
 m_B &\leftarrow 1 \| ID_B \| g^{X_B} \| N_B .
 \end{aligned}$$

Here, 0 and 1 are two public (and fixed) values that are used to break the symmetry and thus prevent a *reflection attack* [79]. Finally,  $ID_A$  and  $ID_B$  are human readable identifiers belonging to parties  $A$  and  $B$  (e.g., e-mail addresses).

The following four messages are exchanged over an insecure radio link. In the first message,  $A$  sends to  $B$  the commitment  $c_A$ .  $B$  responds with his own commitment  $c_B$ . In turn,  $A$  sends out  $d_A$ , by which  $A$  opens the commitment  $c_A$ .  $B$  checks the correctness of the commitment/opening

pair  $(\widehat{c}_A, \widehat{d}_A)$  and verifies that 0 appears at the beginning of  $\widehat{m}_A$ . If the verification is successful,  $B$  sends, in the fourth message,  $d_B$ , by which  $B$  opens the commitment  $c_B$ .  $A$  in turn checks the commitment and verifies that 1 appears at the beginning of  $\widehat{m}_B$ . If this verification is successful,  $A$  and  $B$  proceed to the final phase (Figure 3.6).

In the final phase,  $A$  and  $B$  first generate the verification strings  $s_A$  and  $s_B$ , respectively, as shown on Figure 3.6. The length of each of these strings is  $k$ . Finally, Alice and Bob (as users) simply compare  $s_A$  and  $s_B$  (using visual or vocal communication). If they match, Alice and Bob accept each other's DH public parameters  $g^{X_A}$  and  $g^{X_B}$  and the corresponding identifiers  $ID_A$  and  $ID_B$  as being authentic. At this stage, Alice and Bob can safely generate the corresponding secret DH key ( $g^{X_A X_B}$ ).

We now assess the security of the optimal DH-SC protocol.

**Definition 7** *We say that any protocol  $\Pi(k, (A, B))$  is a secure protocol enabling authentication of DH public parameters between  $A$  and  $B$  if the (polynomial-time) attacker  $M$  cannot succeed in deceiving  $A$  and  $B$  into accepting DH public parameters different than  $g^{X_A}$  and  $g^{X_B}$ , except with a satisfactorily small probability  $O(2^{-k})$ .*

To state the result about the security of DH-SC protocol, we need two additional security parameters ( $k$  was already introduced before: it is the length of verification strings  $s_A$  and  $s_B$ ). As in the case of the MT-SC authenticator, we denote with  $\gamma$  the maximum number of sessions (successful or abortive) of the DH-SC protocol that any party can participate in. We further assume that there are  $n$  parties that are using the DH-SC protocol.

**Proposition 2** *The probability that an attacker succeeds against the DH-SC (with the  $\text{commit}(\cdot)$  being perfectly hiding and computational binding) is bounded by  $n\gamma 2^{-k} + \varepsilon(\text{par})$ , where  $\varepsilon(\text{par}) = P[B]$  is negligible in the security parameter(s)  $\text{par}$  of the used commitment scheme. Therefore, for the appropriately chosen parameter  $k$ , DH-SC is a secure protocol enabling authentication of DH public parameters.*

**Remark 1** *The probability of success by the attacker as stated in Proposition 2, refers to the success against any one among all DH-SC protocol runs (successful or abortive); in other words, the attacker does not care which parties communication he breaks/influences. On the contrary, the probability that the attacker is successful against a fixed (targeted) party is bounded by  $\gamma 2^{-k} + \varepsilon(\text{par})$ .*

We prove Proposition 2 by showing that the DH-SC protocol (Figure 3.6) is just a compact version of the secure DH key agreement protocol obtained by the straightforward application of the MT-SC authenticator to basic DH protocol (Figure 3.5). In this direction, it is sufficient to observe that in the DH-SC protocol, we simply piggyback the messages of one MT-SC authenticator on the other (Figure 3.6). In this case, the  $k$ -bit random string  $N_A$  (sent through the commit/opening pair  $(c_A, d_A)$ ) plays two roles:

- (i) The role of  $N_{A1}$  in the first MT-SC authenticator where Alice wants to send  $m_A = g^{X_A}$  to Bob on Figure 3.5.
- (ii) The role of  $N_{A2}$  in the MT-SC authenticator where Bob wants to send  $m_B = g^{X_B}$  to Alice on Figure 3.5.

The first role is clearly fulfilled by  $N_A$  from the DH-SC protocol on Figure 3.6. That  $N_A$  fulfills the second role follows from the fact that  $N_A$  on Figure 3.6 remains hidden until Alice opens  $c_A$  by

sending out  $d_A$ , which Alice does only after receiving  $\widehat{c}_B$  from Bob (or the adversary). The same analysis is valid for Bob from Figure 3.6. Finally, to break the symmetry of the DH-SC protocol we append two public (fixed) values 0 and 1 to messages  $m_A$  and  $m_B$ , respectively, and impose the verification of both at the end of the protocol on Figure 3.6. Therefore, indeed, the DH-SC protocol can be seen as a compact version of the secure protocol from Figure 3.5, that is, it can be seen as two runs of the MT-SC authenticator (Figure 3.3). With this, we conclude the proof of Proposition 2.

A more formal approach to proving Proposition 2 would be to show that by having a non-negligible advantage against the DH-SC protocol, the adversary has a non-negligible advantage against the MT-SC authenticator, which would contradict the security of the MT-SC (cf. Theorem 10).

### 3.5 Security Analysis of the MT-Authenticator

We analyze the security of the MT-SC authenticator in the Bellare and Rogaway communication model based on *matching conversations* [17].

#### 3.5.1 Matching Conversations

In this model, a protocol  $\Pi(k, I)$  is executed by a pair of parties  $(A, B) \in I$ , where  $I$  is a set of parties that share some common context (e.g., they all run a message authentication protocol). By  $\Pi_{B,A}^t$  we mean that a party  $B$  attempts to authenticate a message from party  $A$  in a session that  $B$  believes has the session identifier  $t \in \mathbb{N}$ . Here, by authentication of a message we mean that at the end of a successful run of the protocol, party  $B$  accepts that a message  $m$  it has received must have been sent by party  $A$ , except with a negligible probability.

We consider an active attacker Mallory in the communication model of Bellare and Rogaway [17], meaning that Mallory can observe, modify and schedule communication between a pair of parties  $(A, B)$ . Given that Mallory is a powerful attacker, we let Mallory interact with  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  as oracles in a “black box” style, meaning that Mallory can query  $\Pi_{A,B}^s$  by supplying  $A$  with input queries that comply to the observed authentication protocol. In the response to any query, oracle  $\Pi_{A,B}^s$  outputs a message that complies to the authentication protocol. We use the following format  $(A, B, s, conv)$  to record all queries and responses that  $\Pi_{A,B}^s$  sent out in the session that  $A$  marks as  $s \in \mathbb{N}$ ; we do “the same” for  $\Pi_{B,A}^t$ . Here,  $conv$  denotes a conversation of  $\Pi_{A,B}^s$ , meaning a sequence of timely ordered messages that  $\Pi_{A,B}^s$  has sent out and received. We say that  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  have matching conversations, if for each message  $m$  sent out by  $\Pi_{A,B}^s$  in time  $\tau_i$ ,  $\Pi_{B,A}^t$  received the same message  $m$  in  $\tau_{i+1}$  and if for each message  $m$  sent out by  $\Pi_{B,A}^t$  in time  $\tau_i$ ,  $\Pi_{A,B}^s$  received the same message  $m$  in  $\tau_{i+1}$  [17]. Here,  $\tau_0 < \tau_1 < \tau_2 < \dots < \tau_R$  is, for some positive integer  $R$ , a time sequence recorded by  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  when conversing.

Consider a pair of oracles  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  that belong to party  $A$  and party  $B$ , respectively. Following the unfolding of the protocol on Figure 3.3, the conversations of  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  can be written as follows:

$$\begin{aligned} conv_A &= (\tau_0, \perp, c), (\tau_2, \widehat{N}_B, d), (\tau_4, \perp, s_A) \\ conv_B &= (\tau_1, \widehat{c}, N_B), (\tau_3, \widehat{d}, \perp), (\tau_5, s_A, \perp), \end{aligned} \tag{3.2}$$

where  $\perp$  means that a party receives/sends no message in the corresponding time  $\tau_i$ . We first observe that if the two conversations are not modified by adversary  $M$ ,  $\Pi_{B,A}^t$  (and hence  $\Pi_{A,B}^s$ ) will



reach the “Accept” decision and  $conv_A$  and  $conv_B$  will be matching. This is obvious because then  $c = \hat{c}$ ,  $d = \hat{d}$  (which implies  $N_A = \hat{N}_A$ ) and  $N_B = \hat{N}_B$ , and therefore  $s_A$  matches  $s_B \leftarrow N_B \oplus \hat{N}_A$ , meaning that  $\Pi_{B,A}^t$  will output “Accept” (Figure 3.3). Moreover,

$$\tau_0 < \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5 .$$

This essentially means that party  $B$  will believe that the message  $\hat{m}$  was sent by party  $A$ .

In Definition 5, we use the term “win” to define the security of any MT-authenticator. In the model of matching conversations, we can define more formally the meaning of the term “win”.

**Definition 8** *We say that  $\Pi(k, (A, B))$  is a secure message authentication protocol between  $A$  and  $B$  if attacker  $M$  cannot win, except with a satisfactorily small probability  $O(2^{-k})$ . Here,  $M$  wins if  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  reach the “Accept” decision while they do not have matching conversations.*

Observe that if any of  $\hat{c}$ ,  $\hat{N}_B$ ,  $\hat{d}$  or  $s_A$  are missing or are received out of order,  $\Pi_{A,B}^t$  and  $\Pi_{B,A}^t$  will simply “Abort” the protocol  $\Pi(k, I)$  and adversary  $M$  will certainly fail to convince  $\Pi_{B,A}^t$  and  $\Pi_{A,B}^s$  to “Accept”.

### 3.5.2 Security of the MT-SC Authenticator

We denote the MT-SC authenticator (cf. Figure 3.3) as a protocol  $\Pi(k, I)$ . We observe a pair of parties  $(A, B) \in I$  running  $\Pi(k, I)$  and a powerful polynomially-bounded active adversary  $M$ .

In our security proof of  $\Pi(k, I)$ , we calculate the probability of the event  $S|\bar{B}$ ; the probability that the adversary is successful against the given user conditioned on the event  $(\bar{B})$  that the binding property of the employed (perfectly hiding and computational binding) commitment scheme holds (see Section 3.3.2). We assume that each party has access to a perfect random number generator. Note that we will observe the security of  $\Pi(k, I)$  in the sense of Definition 8. Let  $\gamma$  be the maximum number of sessions (successful or abortive) that any party can participate in. We will assume that there are at most  $n$  parties using protocol  $\Pi(k, I)$ . In our analysis, we will also assume that each party participates in at most one message authentication session at a time.

We show that for fixed  $(A, B)$  and  $t$ , the probability that oracle  $\Pi_{B,A}^t$  outputs “Accept” without a matching conversation is satisfactorily small. Note that if  $\Pi_{B,A}^t$  outputs “Accept” then there must exist some oracle  $\Pi_{A,B}^s$  (with party  $A$ ) that outputs “Accept” too; message  $s_A$ , at the end of protocol  $\Pi(k, I)$ , guarantees this. We first state the following intuitive result:

**Lemma 3** *If adversary  $M$  is to succeed against a pair of oracles  $(\Pi_{A,B}^s, \Pi_{B,A}^t)$ , then we must have  $c \neq \hat{c}$ , where  $c$  is the commitment sent out by  $\Pi_{A,B}^s$  and  $\hat{c}$  is the commitment received by  $\Pi_{B,A}^t$ .*

*Proof:* Claim: If  $c = \hat{c}$  and  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  both “Accept”, then  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  must have matching conversations. Indeed, the fact that  $\Pi_{B,A}^t$  received a valid commitment opening value  $\hat{d} \neq d$  would contradict the event  $\bar{B}$ , so we must have  $d = \hat{d}$  and hence  $m = \hat{m}$  and  $N_A = \hat{N}_A$ . Furthermore, since  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  both “Accept”, we have  $N_A \oplus \hat{N}_B = N_B \oplus \hat{N}_A$  and hence  $N_B = \hat{N}_B$ . Moreover,  $\tau_0 < \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5$ . Therefore,  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  have matching conversations.  $\square$

If  $\Pi_{B,A}^t$  is to output “Accept”, then the pair  $(\hat{c}, \hat{d})$  has to be a valid commit/opening pair. Furthermore, if oracle  $\Pi_{B,A}^t$  is to output “Accept”, then there must exist some  $\Pi_{A,B}^s$  (with party  $A$ ) that outputs  $s_A \leftarrow N_A \oplus \hat{N}_B$  such that  $s_A = s_B \leftarrow N_B \oplus \hat{N}_A$ . Note here that  $\hat{N}_A$  and  $\hat{N}_B$  are potentially chosen by the adversary  $M$ .

Consider now the interaction between a pair of oracles  $(\Pi_{A,B}^s, \Pi_{B,A}^t)$  and adversary  $M$  as given in (3.2). Assume that  $(\hat{c}, \hat{d})$  is a valid commit/opening pair and assume  $c \neq \hat{c}$  (Lemma 3). Note that if any of the two assumptions does not hold, then  $M$  certainly fails. Then, we can prove the following theorem equivalent to Theorem 9.

**Theorem 11** *For any such interaction between  $\Pi_{A,B}^s$  and  $\Pi_{B,A}^t$  and adversary  $M$ , we have*

$$P[N_A \oplus \hat{N}_B = N_B \oplus \hat{N}_A | \overline{B}] \leq \gamma 2^{-k} .$$

Moreover,  $P[S | \overline{B}] \leq \gamma 2^{-k}$ .

*Proof:* Observe first that  $M$  has to submit  $\hat{N}_B$  before actually seeing  $N_A$ . This follows from the unfolding of  $\Pi(k, I)$  and the (perfectly) hiding property of the commitment scheme (see Figure 3.3 and conversations  $conv_A$  and  $conv_B$  given in (3.2));  $\hat{N}_B$  is received by  $\Pi_{A,B}^s$  at time  $\tau_2$  and only then  $\Pi_{A,B}^s$  sends out (discloses through  $d$ )  $N_A$ . Similarly,  $M$  has to submit  $\hat{N}_A$  (as a part of commitment  $\hat{c}$ ) before actually seeing  $N_B$ . This follows from the unfolding of  $\Pi(k, I)$  and the binding property of the commitment scheme ( $M$  does not change  $\hat{N}_A$  subsequently, since this would contradict the event  $\overline{B}$ );  $\hat{c}$  (commitment to  $\hat{N}_A$ ) is received by  $\Pi_{B,A}^t$  at time  $\tau_1$  and only then  $\Pi_{B,A}^t$  sends out  $N_B$  (see conversations (3.2)).

Thus, irrespectively of the attacking strategy taken by  $M$ , conditioned on the event  $\overline{B}$  (i.e., the binding property holds), either  $N_A$  or  $N_B$  will be disclosed after  $\hat{N}_A$  and  $\hat{N}_B$  have been generated and submitted. If it happens that both  $N_A$  and  $N_B$  are disclosed at the exactly same time, then we pick an arbitrary one.

Assume that  $N_A$  is disclosed after  $N_B$ . Then, we have

$$P[N_A \oplus \hat{N}_B = N_B \oplus \hat{N}_A | \overline{B}] = P[N_A = N_B \oplus \hat{N}_A \oplus \hat{N}_B | \overline{B}] \leq \gamma 2^{-k} ,$$

that is, (1)  $N_A$  and  $N_B$  are independent and uniformly distributed random variables, (2)  $\hat{N}_A$  and  $\hat{N}_B$  are both generated and submitted before  $N_A$  is disclosed (therefore,  $\hat{N}_A$  and  $\hat{N}_B$  are also independent of  $N_A$ ), and (3) by assumption, party  $A$  can participate in at most  $\gamma$  sessions. The same holds for the case where  $N_B$  is disclosed after  $N_A$ .

Since the condition  $(N_A \oplus \hat{N}_B = N_B \oplus \hat{N}_A)$  is the necessary condition for the attacker to be successful (to win), we must have  $P[S | \overline{B}] \leq \gamma 2^{-k}$ .

We conclude the proof by observing that the assumption  $c \neq \hat{c}$  precludes from trivial situations, where  $M$  would not modify the messages, to take place; in which case we would have  $P[N_A \oplus \hat{N}_B = N_B \oplus \hat{N}_A | \overline{B}] = 1$ .  $\square$

From Theorem 11, we conclude that the probability that there exists oracle  $\Pi_{B,A}^t$  that belongs to party  $B$  and that “Accepts” without a matching conversation is at most  $2^{-k}$  times the maximum number of interactions (successful or abortive) that party  $B$  has participated in. It is crucial that we take abortive attempts into account, too, when evaluating the probability that  $M$  is successful against a given party. This is because  $M$  learns that his attempt is unsuccessful (i.e.,  $N_A \oplus \hat{N}_B \neq N_B \oplus \hat{N}_A$ ) before  $M$  potentially sends out  $\hat{d}$  in an attempt to disclose  $\hat{N}_A$  to party  $B$ . If  $M$  is not successful in a given attempt, he can simply abort the protocol by simply not sending  $\hat{d}$  to  $B$ .

Note that party  $A$  “Accepts” only if the corresponding party  $B$  “Accepts”. Therefore, the probability that there exists oracle  $\Pi_{A,B}^s$  that belongs to party  $A$  and that “Accepts” without a matching conversation is at most  $\gamma 2^{-k}$ . Finally, the probability that any party is broken, assuming that there are  $n$  parties that use protocol  $\Pi(k, I)$ , is at most  $n\gamma 2^{-k}$  (cf. Proposition 2).

### 3.6 Diffie-Hellman Key Agreement Based on Distance Bounding (DH-DB)

In this section, we describe a key agreement protocol that is based on verifiable principal proximity, achieved through distance bounding. We call our protocol Diffie-Hellman with Distance-Bounding (DH-DB). The protocol ensures the secure establishment of a shared key between two parties  $A$  and  $B$  if there are no other parties that are closer to  $A$  or to  $B$  than they are to each other. In this section, we assume that the pair of devices have the means to accurately estimate the distance between themselves (later in this section we discuss the possible techniques for this purpose).

The proximity check between the two devices is performed through distance bounding [23]: each device upper-bounds its distance to the device with which it is agreeing on a key. The measured distance appears on both device displays. The users then visually check whether there are other users/devices closer to them than the displayed distance bounds. If this is not the case, the exchanged DH public parameters and the corresponding identities are accepted.

The DH-DB protocol is shown on Figure 3.7. Note that the protocol on Figure 3.7 is actually built upon the DH-SC protocol (Figure 3.6). The only difference is that the verification of the authentication strings  $s_A$  and  $s_B$  (in the DH-DB protocol) is performed through Brands and Chaum's distance bounding protocol [23]. Thus, Alice ( $A$ ) and Bob ( $B$ ) exchange the commitment/opening pairs  $(c_A, d_A)$  and  $(c_B, d_B)$  in the first four messages in exactly the same way as in DH-SC protocol. Furthermore,  $A$  and  $B$  perform all necessary verifications as in the DH-SC protocol. Finally,  $A$  and  $B$  calculate  $k$ -bit verification strings  $s_A$  and  $s_B$ . As we can see on Figure 3.7,  $A$  and  $B$  also exchange commitments  $c'_A$  and  $c'_B$  to concatenations  $0\|R_A$  and  $1\|R_B$ ; again, 0 and 1 serve to protect against the reflection attack.

Upon reception of the commitments  $c'_A$  and  $c'_B$ , the devices execute distance bounding by exchanging bit by bit all the bits of  $R_A$ ,  $R_B$ ,  $s_A$  and  $s_B$  as shown on Figure 3.7. During distance bounding, the devices measure round-trip times between sending a bit and receiving a response bit. The device estimates the distance-bound to the other device by multiplying the round trip time by the speed of light in the case of the radio or by the speed of sound in the case of ultrasound communication.

Having exchanged  $R_A$ ,  $R_B$ ,  $s_A$  and  $s_B$ ,  $A$  and  $B$  open  $c'_A$  and  $c'_B$  by sending out  $d'_A$  and  $d'_B$ , which they then use to retrieve  $\widehat{R}_B$  and  $\widehat{R}_A$ , respectively.  $A$  and  $B$  then use  $\widehat{R}_B$  and  $\widehat{R}_A$  to retrieve  $\widehat{s}_B$  and  $\widehat{s}_A$ ; this is done by performing a series of  $k$  "xor" operations as shown on Figure 3.7. Finally,  $A$  and  $B$  verify  $\widehat{s}_B$  and  $\widehat{s}_A$  against  $s_A$  and  $s_B$ ; note that this verification is now done by the devices  $A$  and  $B$ , whereas in the DH-SC protocol this comparison is performed by users  $A$  and  $B$ .

Having successfully verified  $\widehat{s}_B$  against  $s_A$  and  $\widehat{s}_A$  against  $s_B$ , the devices  $A$  and  $B$  display the measured distance bounds on their screens. The users  $A$  and  $B$  then visually verify that there are no other users/devices in their vicinity (in what we call the *integrity region* of  $A$  and  $B$ ; see Figure 3.8). If the displayed distance bound corresponds to the distance to the closest device, the users accept the exchanged DH public parameters  $g^{X_A}$  and  $g^{X_B}$  and the corresponding identities  $ID_A$  and  $ID_B$  as being authentic; otherwise, they reject them. This last step is important as it guarantees that the exchanged messages in the protocol preserved their integrity, meaning that they cannot have been maliciously modified or generated by an adversary, but only by the closest party.

#### 3.6.1 Properties of DH-DB Protocol

In DH-DB, the MITM attack is prevented by the proximity verification. We define the *integrity region* of users  $A$  and  $B$  as the union of two spheres each centered at the position of devices  $A$  and

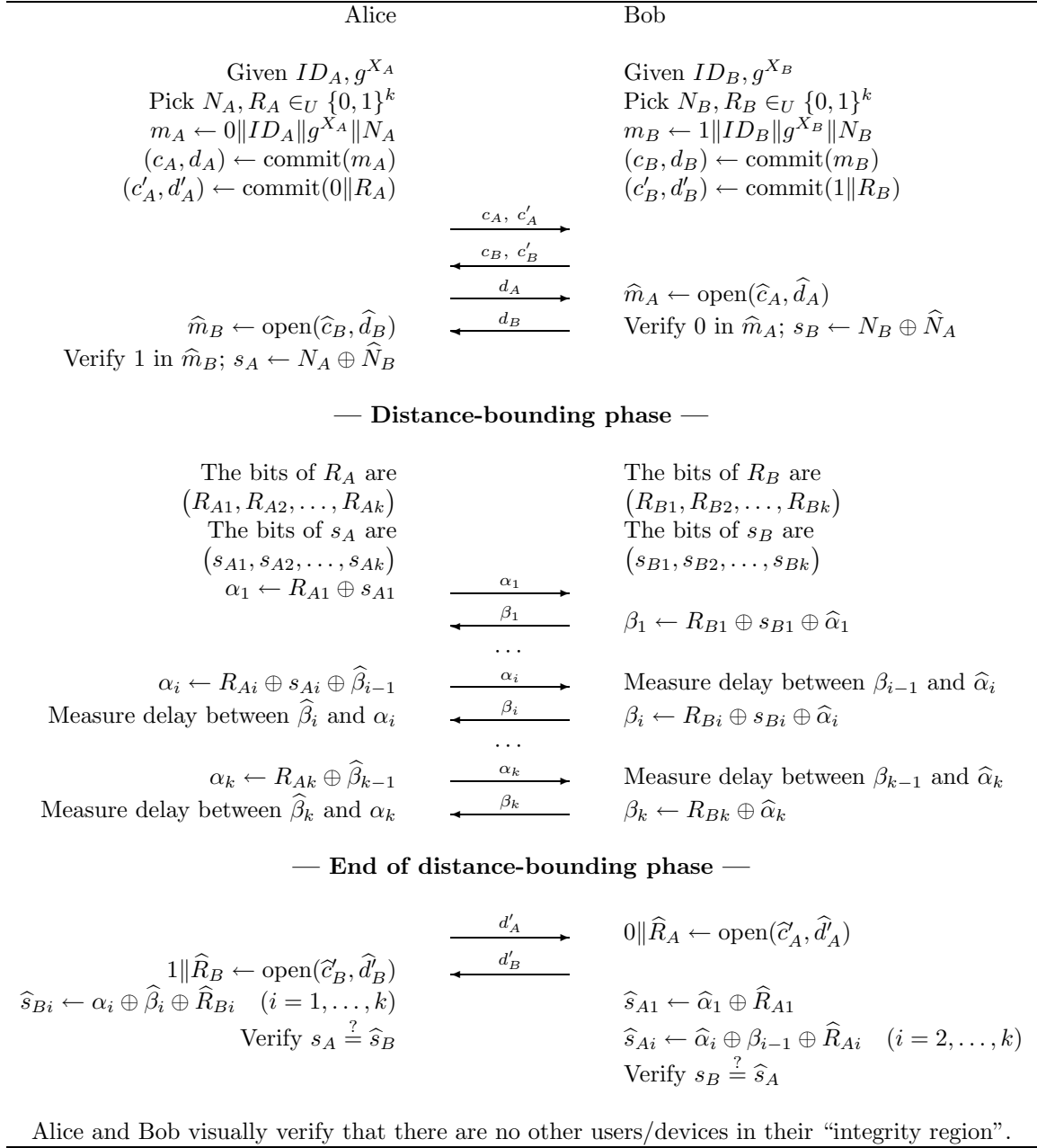


Figure 3.7: Operation of the Diffie-Hellman key agreement with Distance Bounding; all the communication takes place over an insecure (high-bandwidth) channel.

$B$  with radii equal to the distance  $d$  between devices  $A$  and  $B$  (see Figure 3.8). If the users can visually verify that there are no other users/devices within the integrity region and if the distance-bounding phase is secure, then the integrity of messages  $s_A$  and  $s_B$  is respected; i.e.,  $s_A$  and  $s_B$  sent from  $A$  and  $B$  will reach  $B$  and  $A$ , respectively, unchanged. Note here that the security of the distance-bounding phase relies on the fact that the attacker does not learn  $R_A$  and/or  $R_B$  until the end of this phase; all that  $M$  knows are commitments  $c'_A$  and  $c'_B$ . Therefore,  $R_A$  and  $R_B$  guarantee

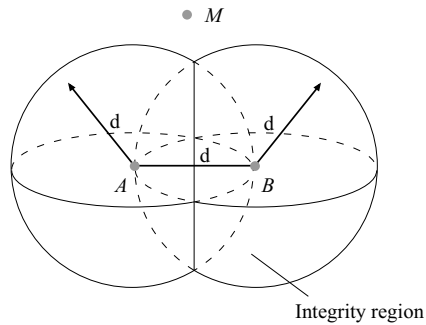


Figure 3.8: Integrity region of users  $A$  and  $B$  ( $d$  is the distance between users’ devices)

to  $A$  and  $B$  that the attacker cannot send the bits, in the distance-bounding phase, earlier than receiving the previous bit; for this reason, it cannot appear to be closer than it actually is.

If attacker  $M$  is not within the integrity region, he will not be able to send messages to  $A$  such that it seems that it is placed on the same (or shorter) distance from  $A$  as  $B$ . With this, the integrity of  $s_A$  and  $s_B$  is preserved as if users  $A$  and  $B$  exchanged  $s_A$  and  $s_B$  face to face (e.g., voice communication). Since  $s_A$  and  $s_B$  are actually authentication strings from the DH-SC protocol, by verifying that  $s_A$  and  $s_B$  match, users  $A$  and  $B$  are guaranteed that messages  $m_A$  and  $m_B$  are authentic, except with a satisfactorily small probability (see Proposition 2).

A nice property of this protocol is that it does not depend on the power ranges of the devices, but solely on their proximity  $d$ . Specifically, the closer the parties are, the smaller the integrity region is, and the harder it is for the adversary to get into the region without being noticed by the honest parties. For example, getting the devices as close as  $d = 30$  cm should be a sufficient guarantee, even for the most demanding users, that no adversary (be it even a small device) resides in the corresponding integrity region.

### 3.6.2 Implementation

We envision two possible implementations of DH-DB: with radio (RF) and with ultrasound (US). Both exhibit equal security guarantees, but require different equipment attached to the devices. We briefly report on how these implementations have been addressed so far. Brands and Chaum [23] propose a distance bounding protocol that can be used to verify the proximity of two devices connected by a radio link; it requires devices with a high (nanosecond) precision-of-time measurement. To the best of our knowledge, the only commercial technique that achieves such precision, and achieves therefore a high precision-of-distance measurement, is Ultra Wide Band (UWB). In [38], Fontana has demonstrated that with UWB, distances can be measured with an error margin of up to 15 cm.

Sastry, Shankar and Wagner [93] propose a distance bounding protocol based on ultrasound and radio wireless communication (a similar technique was also proposed by Waters and Felten [106]). Ultrasound-based distance bounding requires only millisecond time measurement precision, but of course it needs each device to be able to communicate via ultrasound. Ultrasound-based distance bounding has centimeter precision.

In both radio-frequency and ultra-sound solutions, the response time (the “xor” operation and the reversion of the transceiver) of the challenged principal must be tightly bound and predictable.

### 3.7 Related Work

The problem of key establishment is a very active area of research. Stajano and Anderson propose the *resurrecting duckling* security policy model, [102] and [101], in which key establishment is based on the physical contact between communicating parties (their PDAs). A physical contact acts as a *location limited channel*, which can be used to transmit a key (or a secret) in plaintext. Thus, no cryptography is required at this stage. The potential drawback of this approach is that the realization of a physical contact can be cumbersome with bulky devices (e.g., laptops).

An approach inspired by the resurrecting duckling security policy model is proposed by Balfanz et al. [14]. In this work, the authors go one step further and relax the requirement that the location limited channel has to be secure against passive eavesdropping; they introduce the notion of a *location-limited channel* (e.g., an infrared link). A location-limited channel is used to exchange pre-authentication data and should be resistant to active attacks (e.g., man-in-the-middle). Once pre-authentication data are exchanged over a location-limited channel, users switch to a common radio channel and run any standard key exchange protocol over it. Possible candidates for a location-limited channel include: physical contact, infrared, and sound (ultrasound) [14]. Here again, the disadvantage of this approach is that it may be a cumbersome to realize a link with bulky devices (e.g., laptops) in the case of infrared or physical contact. In addition, the infrared link itself is not well studied in the context of secure communications. Actually, our DH-SC protocol could be applied to the infrared link as well.

Asokan and Ginzboorg propose another solution based on a shared password [13]. They consider the problem of setting up a session key between a group of people (i.e., their computers) who get together in a meeting room and who share no prior context. It is assumed that they do not have access to public key infrastructure or third party key management services. The proposed solution is the following. A fresh password is chosen and shared among those present in the room (e.g., by writing it on a sheet of paper or a blackboard). The shared password is then used to derive a strong shared session key. This approach requires users to type the chosen password into their personal devices.

It is well known that IT security systems are only as secure as their weakest link. In most IT systems the weakest links are the users themselves. People are slow and unreliable when dealing with meaningless strings, and they have difficulties remembering strong passwords. In [86], Perrig and Song suggest using hash visualization to improve the security of such systems. Hash visualization is a technique that replaces meaningless strings with structured images. However, having to compare complex images can be cumbersome.

In US patent no. 5,450,493 [77], Maher presents several methods to verify DH public parameters exchanged between users. The first method described in [77] is the most relevant one for the problem we consider in this chapter; other methods are based on certificates and/or shared secrets. Thus,  $A$  and  $B$  first perform the DH key exchange protocol and in turn report to each other values  $f(K_A)$  and  $f(K_B)$ , where  $K_A$  and  $K_B$  are the shared DH keys as computed by  $A$  and  $B$ , respectively, and  $f$  is a compression function (i.e.,  $f$  maps a key to 4-digit hex vectors [77]). Unfortunately, this technique has a flaw, which was discovered by Jakobsson [51]. The problem with Maher's technique is the following. An attacker Mallory  $M$ , who knows  $f$  and controls all the communication, first generates his secret exponents  $X_1$  and  $X_2$  and the corresponding public parameters  $g^{X_1}$  and  $g^{X_2}$ . Since  $M$  knows that  $A$  and  $B$  will compare  $f(g^{X_A X_2})$  and  $f(g^{X_B X_1})$ , he checks if  $f(g^{X_A X_2}) = f(g^{X_B X_1})$ . If this is the case,  $M$  sends  $g^{X_2}$  instead of  $g^{X_B}$  to  $A$ , and  $g^{X_1}$  instead of  $g^{X_A}$  to  $B$ . If  $f(g^{X_A X_2}) \neq f(g^{X_B X_1})$ ,  $M$  generates new values for  $X_1$  and  $X_2$  and repeats the above procedure. Since  $f$  outputs a very short string (4-digit hex vector [77]),  $M$  will find a

collision after a relatively low number of attempts.

Motivated by the flaw in [77], Jakobsson [51] and Larsson [66] proposed two solutions. However, both solutions are based on a temporary secret shared between the two users (thus, for example, SHAKE stands for *Shared key Authenticated Key Exchange*). In our work, we consider the same problem but in a more demanding setting, as we assume that the users share no secret key prior to the key exchange.

Dohrmann and Ellison [33] propose a method for key verification that is similar to our approach; this method is based on converting key hashes to readable words or to an appropriate graphical representation. However, it seems that users are required to compare a substantial number of words (or graphical objects); this task could take them as much as 24 seconds according to [33]. This time is significantly reduced when the graphical representation is used. However, Dohrmann and Ellison provide no security analysis of their approach.

In [43] and [44], Gehrman et. al., propose a set of techniques to enable wireless devices to authenticate one another via an insecure wireless channel with the aid of the manual transfer of data between the devices. The protocol, which they call MANA II, is similar to our DH-SC protocol; in both protocols the parties have to compare the output of their devices. The MANA II protocol is based on authentication codes. At the end of the protocol the parties have to compare a key and a check value, where only the check value contributes to the uncertainty of the attacker. As a result, with MANA II the number of bits to be compared by the parties is twice as much as with our DH-SC. Other mechanisms proposed by [43] and [44] basically require the users to type in given values into their devices. The important difference between MANA II and our DH-SC protocol is that MANA II requires the parties to compare two strings (a key and a check value), whereas only one string (the check value) contributes to the uncertainty of the attacker. As a result, for a fixed security level of, MANA II requires the parties to compare twice as many bits as in the case of the DH-SC protocol.

In [11], Alpern and Schneider [11] present a protocol that allows two parties to agree on a secret key on channels for which an adversary cannot tell who is the source of each message. It is a pairing scheme that does not rely on public-key cryptography. As a follow-up, in [27], Castelluccia and Mutaf propose two movement-based pairing protocols for CPU-constrained devices. Unfortunately, we discovered serious security flaws in the proposed protocols; for example, contrarily to the claim of [27], one of the protocols leaks all information about an agreed shared key.

We should mention other key-exchange protocols, proposed primarily for the use in the Internet: IKE [5], JFK [8] and SIGMA [61]. All these protocols involve authentication by means of digital signatures, which clearly does not fit the problem we study here. We also should mention the work of Corner and Noble [32], who consider the problem of transient authentication between a user and his device, as well as the work of Čapkun et. al [26], where the authors show how to make use of users mobility to bootstrap secure communication in open ad hoc networks.

Finally, we acknowledge the contribution of Perrig et. al. in [85], where the authors propose Tesla, a protocol for broadcast authentication based on delayed key disclosure.

### 3.8 Summary

In this chapter, we have provided two solutions to the fundamental problem of key agreement over an insecure (radio) link. As user-friendliness is extremely important for the acceptance of any security scheme, we have minimized the burden on the user: there is no need for physical contact, nor for infrared communication between the devices.

We have proposed a new (re-usable) MT-authenticator (MT-SC) based on string comparison, by which users can optimally trade-off the desired security with their involvement in the protocol execution. It is also shown, how the MT-SC authenticator can be used in a modular way to build secure key agreement protocols in the setting where users share no prior secret or certified information. All users have to do is to compare a short authentication string.

Finally, we have introduced the notion of integrity region in the context of distance bounding based Diffie-Hellman key agreement protocol. To the best of our knowledge, this is a novel security property<sup>3</sup>.

---

<sup>3</sup>In the following chapter, we introduce another novel security property called *authentication through presence*.



## Chapter 4

# Integrity ( $I$ ) codes for Message Integrity Protection Over Insecure Channels

### 4.1 Introduction

Conventional security goals such as message confidentiality, integrity, and authentication are traditionally achieved through the use of certified public-keys or shared secret keys, and by the application of appropriate cryptographic primitives (i.e., encryption schemes, signatures, message authentication codes, etc.).

Similarly to Chapter 3, in this chapter, we propose *integrity codes* ( $I$ -codes), a new security primitive (mechanism) that enables integrity protection of the messages exchanged between entities that do not hold any shared secrets or mutual authentication material (i.e. public keys or shared secret keys). The construction of  $I$ -codes enables a sender to encode any message, such that if its integrity is violated in transmission, the receiver is able to detect it. In the literature such codes are known as *All-Unidirectional Error-Detecting* codes and are used in situations where it is possible to change a bit “0” into a bit “1” but the contrary is not possible (except with a negligible probability) [18, 21, 20]. An all-unidirectional error-detecting code is able to detect any number of unidirectional errors (i.e., “0”  $\rightarrow$  “1”) in the given codeword; in other words, for the given error-detection code no unidirectional error can transform a (valid) codeword into another (valid) codeword. Unidirectional error-detecting codes find application, for example, in the encoding of unchangeable data on digital optical disks [67].

Our main goal in this study is to propose a mechanism to protect the integrity of messages exchanged between entities in the presence of an adversary who tries to convince the entities to accept modified messages as being authentic. We do not attempt to increase reliability of message transmissions – actually, as we will see shortly, we will have to sacrifice the reliability of message transfer in order to achieve our goal. For these reasons, we find it more appropriate to call the error-detecting codes simply *integrity codes* ( $I$ -codes).

Our approach to message integrity protection involves three main components: *on-off keying*, *signal anti-blocking* and *I-coding*. On-off keying is a modulation by which the bit “1” is transmitted on the channel as the *presence* of a signal and the bit “0” is transmitted as the *absence* of a signal. Signal anti-blocking means that the energy of the signal (bit “1”) cannot be annihilated by an adversary (we show several ways how to possibly ensure this). Finally, by  $I$ -coding we mean that a

message is encoded using  $I$ -codes (described in Section 4.3) before its transmission over an insecure channel.

With these three components, we can ensure that only bits “0” (but not bits “1”) can be flipped by the adversary on the channel and that if a bit is flipped, this will be detected at the receiver, which is guaranteed by the properties of  $I$ -codes (Section 4.3).

We further show how this approach based on  $I$ -codes can be implemented on a radio communication channel. To validate our concept, we implement and test  $I$ -codes, on-off keying and signal anti-blocking components on a Mica2 wireless sensor network platform; our implementation demonstrates that the approach based on  $I$ -codes can be implemented using existing radio and processing hardware and protocols at virtually no extra cost. Ensuring integrity protection over insecure radio channels is particularly important for preventing “man-in-the-middle”-based attacks, which could otherwise be perpetrated on a radio channel. By taking advantage of the characteristics of radio channels, the  $I$ -codes help to completely prevent this attack.

Using  $I$ -codes, we develop a novel concept called *authentication through presence*, which enables message authentication based solely on the awareness of presence in the power range of an entity. We show the application of authentication through presence in two examples: (1) IEEE 802.11 access point authentication, and (2) key establishment over insecure radio channels.

We perform a detailed analysis of the security of  $I$ -codes on a radio channel and we show that they are secure in a realistic attacker model. This analysis takes into account the characteristics of radio channels such as phase shifts, noise, and the attacker’s ability to detect, jam and alter the messages on a channel.

The chapter is organized as follows. In Section 4.2, we state our problem and we describe our system and the attacker model. In Section 4.3, we formally introduce  $I$ -codes and we provide details about their properties. In Section 4.4, we present the results of the  $I$ -codes implementation. In Section 4.5, we show how to use  $I$ -codes for authentication; we introduce the notion of an *authentication through presence*. In Section 4.6, we present the security analysis of  $I$ -codes. Finally, we summarize the chapter in Section 4.8.

## 4.2 Problem Statement and Assumptions

Similarly to Chapter 3, we observe the following problem

*Assuming that two entities ( $A$  and  $B$ ) share a common communication channel (radio channel), but do not share any secrets or authentication material (e.g., shared keys or authenticated public keys), how can the messages exchanged between these entities be authenticated and how can their integrity be preserved in the presence of an attacker ( $M$ )?*

Here, by message integrity, we mean that the message must be protected against any malicious modification, and by message authentication we mean that it should be clear who the sender of the message is.

We assume that the two entities involved in the communication ( $A$  and  $B$ ) do trust each other; otherwise, little can be done. Whenever we speak of the security of a given protocol, we implicitly assume that the entities involved in the protocol are not compromised. We do assume that the entities know (public) protocol parameters.

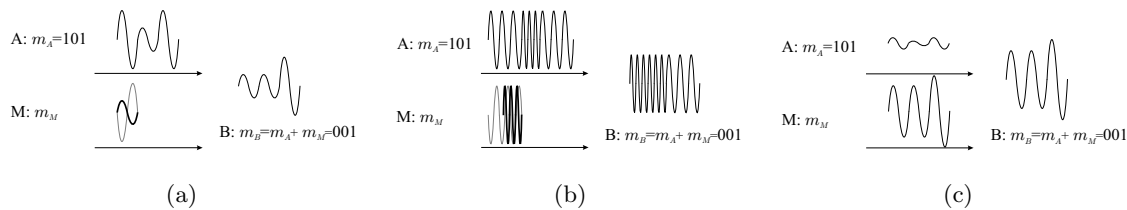


Figure 4.1: Example of attacks on message integrity considered in this chapter: (a) Bit flipping; signals modulated using amplitude modulation (AM); (b) Bit flipping; signals modulated using frequency modulation (FM); (c) Signal overshadowing; signals modulated using amplitude modulation.

### Attacker Model

We adopt the following attacker model. We assume that the attacker Mallory ( $M$ ) controls the communication channel in a sense that he can eavesdrop messages and modify transmitted messages by adding his own messages to the channel. We further assume that the attacker cannot disable the communication channel (e.g., use a Faraday’s cage to block the propagation of radio signals). The attacker can jam the transmission in a way that prevents the transfer of the information contained in the message. However, the receiver will still receive the message from the sender, superimposed by the attacker’s messages. Finally, we assume  $M$  to be *computationally bounded*.

It is interesting to observe that the security of  $I$ -codes themselves does not depend on the attacker being computationally bounded. However, authentication schemes derived from  $I$ -codes presented in Section 4.5 require the attacker to be computationally bounded.

Our attacker model is similar to the the Dolev-Yao model in that the attacker controls the communication channel, but it differs in that we assume that the attacker cannot fully schedule message transmission as it cannot disable the communication channel. This means that the attacker cannot trivially remove the energy of the signal from the channel (we discuss this in more detail in Section 4.6).

Before introducing our solution to the above stated problem, we give some examples of attacks on message integrity on the radio channel, which are relevant to our proposal. Figure 4.1 shows two types of such attacks. The first type of attack is called *bit flipping*, in which the attacker introduces a signal on the channel that converts bit “0” into “1” or vice-versa. This attack is shown on Figure 4.1(a) and Figure 4.1(b) for messages modulated using amplitude and frequency modulation, respectively. Here, the bit is flipped such that the attacker adds to the channel the signal of the opposite phase to the one representing the bit and the signal representing the opposite bit. The second type of attack is the signal *overshadowing attack*, shown on Figure 4.1(c). In this attack, the attacker adds to the channel a signal representing a bit string different from the one sent by the honest entity with a significantly higher power than the one of the original signal. In this way, the original signal, regardless of its format or modulation, becomes entirely overshadowed by the attacker’s signal, and is treated as noise by the receiver.

In the following sections, we show how these and similar attacks on message integrity can be detected through the use of  $I$ -codes in conjunction with on-off keying and signal anti-blocking components. Even though we make a clear distinction between  $I$ -codes and on-off keying, that is, signal anti-blocking, we will often abuse the terminology and call the triple ( $I$ -codes, on-off keying, signal anti-blocking) an  $I$ -code.

### 4.3 Integrity ( $I$ )-codes

Similar to a message authentication code (MAC) that involves a shared secret key, and a signature scheme that involves certified public keys, an integrity code ( $I$ -code) provides a method of ensuring the integrity (and a basis for authentication) of a message transmitted over a public channel. The main difference is that an  $I$ -code removes the assumption that the parties involved in the message exchange share any prior secrets or/and certified public keys.

#### 4.3.1 Definition

$I$ -codes allow a receiver  $B$  to verify the integrity of the message received from the sender  $A$ , based solely on message coding. We now give a more formal definition of integrity codes and the terminology we will use.

**Definition 9** *An integrity code is a triple  $(\mathcal{S}, \mathcal{C}, e)$ , where the following conditions are satisfied:*

1.  $\mathcal{S}$  is a finite set of possible source states (plaintext)
2.  $\mathcal{C}$  is a finite set of binary codewords
3.  $e$  is a source encoding rule  $e : \mathcal{S} \rightarrow \mathcal{C}$ , satisfying the following:
  - $e$  is an injective function
  - it is not possible to convert codeword  $c \in \mathcal{C}$  to another codeword  $\hat{c} \in \mathcal{C}$ , such that  $\hat{c} \neq c$ , without changing at least one bit “1” of  $c$  to bit “0”.

To make the above definition more concrete, we now give two examples of  $I$ -codes.

**Example 1 (Complementary encoding, Manchester code)** The encoding rule ( $e$ ) is the following:

$$\begin{aligned} 1 &\longrightarrow 10 \\ 0 &\longrightarrow 01 . \end{aligned}$$

Assume now that we want to encode messages from the set  $\mathcal{S} = \{00, 01, 10, 11\}$  using the above encoding rule. Then,  $\mathcal{C} = \{0101, 0110, 1001, 1010\}$ , i.e.,  $e(00) = 0101$ ,  $e(01) = 0110$ ,  $e(10) = 1001$ , and  $e(11) = 1010$ . This encoding rule is clearly injective. Note further that each codeword  $c \in \mathcal{C}$  is characterized by the equal number of “0”s and “1”s. Therefore, it is not possible to convert one codeword  $c \in \mathcal{C}$  to a different codeword  $\hat{c} \in \mathcal{C}$ , without flipping at least one bit “1” to bit “0”. For example, to convert  $c = 0110$  into  $\hat{c} = 0101$ , the third bit of  $c$  has to be changed to 0. By Definition 9, the triple  $(\mathcal{S}, \mathcal{C}, e)$  is an  $I$ -code.

**Example 2 (Codes with fixed Hamming weight)** We encode each source state  $s \in \mathcal{S}$  into a binary sequence (codeword) of the fixed length ( $\ell$ ) and fixed Hamming weight ( $w$ ). For binary sequences, Hamming weight is the number of bits “1” in the binary sequence. As in the previous example, suppose  $\mathcal{S} = \{00, 01, 10, 11\}$ . Let  $\ell = 4$  and  $w = 3$ . Then the number of possible binary sequences of length  $\ell$  and with Hamming weight  $w$  is  $\binom{\ell}{w} = \binom{4}{3} = 4$ ; i.e.,  $\{0111, 1011, 1101, 1110\}$ .

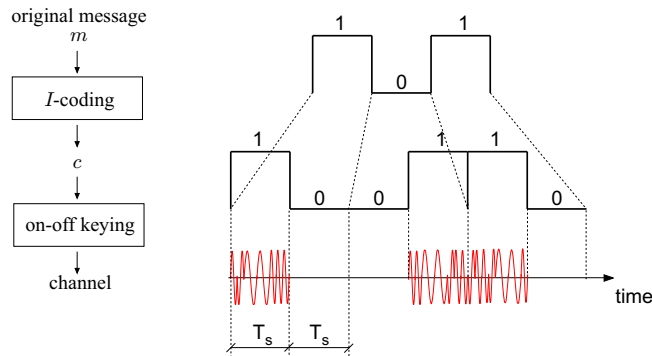


Figure 4.2: An example of  $I$ -coding at the sender using the complementary encoding rule:  $1 \rightarrow 10$  and  $0 \rightarrow 01$ .

Let us define the set of codewords  $\mathcal{C}$  as follows:  $\mathcal{C} \equiv \{0111, 1011, 1101, 1110\}$ . Suppose further the following source encoding rule:

$$\begin{aligned} 00 &\longrightarrow 0111 \\ 01 &\longrightarrow 1011 \\ 10 &\longrightarrow 1101 \\ 11 &\longrightarrow 1110, \end{aligned}$$

that is,  $e(00) = 0111$ ,  $e(01) = 1011$ ,  $e(10) = 1101$  and  $e(11) = 1110$ . Clearly,  $e$  is injective. Moreover, no codeword  $c \in \mathcal{C}$  can be converted into a different codeword  $\hat{c} \in \mathcal{C}$ , without flipping at least one bit “1” of  $c$  to bit “0”. Therefore, by Definition 9, the triple  $(\mathcal{S}, \mathcal{C}, e)$  is an  $I$ -code. It is interesting to observe that the security of the Merkle one-time signature scheme is based on codes with the fixed (known) Hamming weight [80].

In the following section, we show how  $I$ -code can be used on a *radio channel* to ensure the message integrity. However, as we will show,  $I$ -codes are applicable to any communication media (channel) for which we can ensure that it is not possible to block emitted signals on it, except with a negligible probability.

### 4.3.2 $I$ -codes on a Radio Channel

Let us consider a simple example shown on Figure 4.2. Here,  $m$  denotes the message for which the integrity should be checked. Using the given  $I$ -code (i.e., the complementary encoding rule), the sender first encodes  $m$  into the corresponding  $I$ -code codeword  $c$ . Due to the injective property of  $I$ -codes (Definition 9), it is possible to recover unambiguously message  $m$  from the codeword  $c$ . In order to transmit  $c$  over a given radio channel, the sender uses the following *on-off keying* modulation at the physical layer. For each symbol “1” of  $c$ , the sender emits some signal (waveform) during the period  $T_s$  (the *symbol period*). For each symbol “0” of  $c$ , however, the sender emits nothing during period  $T_s$  (Figure 4.2). The waveforms that are transmitted do not carry any information, but it is the *presence* or *absence* of energy in a given time slot of duration  $T_s$  that conveys information<sup>1</sup>.

<sup>1</sup>Note that this is similar to the *pulse position modulation* (PPM).

In order to retrieve the codeword transmitted, the receiver simply measures the energy in the corresponding time slots of duration  $T_s$ . We will assume for the moment that the sender and the receiver are synchronized at the physical layer and with respect of the beginning and the end of the transmission of  $c$ ; later in the chapter, we discuss how this can be achieved. Let  $P_r$  denote the average power that the receiver measures in a given time slot of duration  $T_s$ . Let us also denote with  $P_0$  a pre-defined *threshold power level*. For the given time slot, the receiver decodes the received signals as follows: (1) if  $P_r \geq P_0$ , output symbol “1”, and (2) if  $P_r < P_0$ , output symbol “0”.

In our example on Figure 4.2, the receiver (which is, by assumption, synchronized with the transmitter), listens on the channel during time period  $6 \times T_s$  and for each time slot of duration  $T_s$  it applies the above decoding rule. Finally, the receiver uses the inverse of the used encoding rule (i.e.,  $01 \rightarrow 0$ ,  $10 \rightarrow 1$ ) to retrieve the emitted message  $m = 101$ .

Note that the receiver does not have to know the waveform emitted by the sender. All the receiver has to know is the frequency band used by the sender; the receiver can be thought of as being a bank of radiometers measuring the energy in the given frequency band.

Assume that we can ensure for the used radio channel that it is not possible to block (annihilate) signals emitted over it, except with a negligible probability. Also, the transmitter should transmit signals using a power level high enough so that the average power as measured by the receiver is above the threshold  $P_0$ .

**Theorem 12** *Assuming that the sender and the receiver are synchronized with respect to the beginning and the end of the transmission of the codeword  $c$ , an adversary cannot trick the receiver into accepting the message  $\hat{m}$  when  $m \neq \hat{m}$  is sent, except with a negligible probability.*

*Proof:* From the injective property of the *I*-code (Definition 9) we have

$$\hat{m} \neq m \Rightarrow \hat{c} \neq c ,$$

where  $\hat{c}$  is the unique *I*-code codeword corresponding to message  $\hat{m}$ . Furthermore, converting the codeword  $c$  to another *valid* codeword involves flipping at least one symbol “1” of  $c$  into symbol “0” (Definition 9). Finally, the on-off keying modulation implies that the adversary has to delete (cancel) at least one signal (waveform) emitted on the channel (see Figure 4.2).

However, according to our assumption, the adversary can delete the signal emitted on the used radio channel only with a negligible probability. The need for the synchronization between the sender and the receiver is clear.  $\square$

We note that the adversary can still convert symbol “0” to symbol “1”. In this case, however, the receiver will simply drop the received codeword since such a codeword cannot be demodulated properly. Referring to the example on Figure 4.2, assume that the adversary flips the third symbol “0” into symbol “1” in the original codeword  $c = 100110$ . The receiver will decode the altered codeword as 101110. But this codeword cannot be related to any message, since there is no transformation defined for the pair 11. Therefore, flipping symbol “0” to symbol “1” can be thought of as a DoS attack, which the adversary can mount in any case against a radio channel (no matter which modulation scheme is used).

### 4.3.3 Preventing the Attacker from Erasing Symbol “1”

In order to erase the signal from the channel (symbol “1”), the attacker needs to be able to predict the shape of the signal at the receiver and send the inverted signal to the receiver to cancel it out. There are two major factors that make it difficult for the attacker to erase the signal from

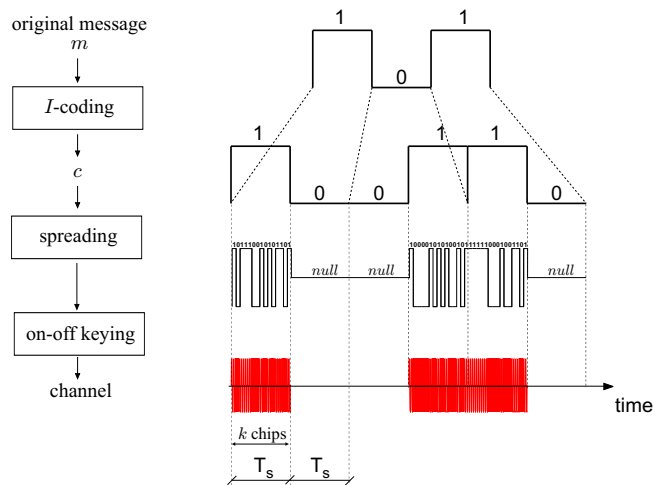


Figure 4.3: An example of  $I$ -coding with spreading using FSK modulation.

the channel: the randomness of the channel and the randomness of the signal generated at the sender. In Section 4.6, we analyze in detail the effects of the randomness of the radio channel on the attacker’s ability to erase the signal from the channel. Here, we focus on the randomness of the signals generated at the sender.

To prevent the attacker from erasing the signal, we implement the following scheme: the sender randomizes the signals corresponding to symbols “1”. It is important to stress that this measure makes sense only if the designated receiver can demodulate the signal at approximately the same speed as the attacker. Specifically, to prevent signal erasure, each symbol “1” of the  $I$ -coded message  $c$  is transmitted as a random signal of duration  $T_s$ . Note that we can randomize amplitude, phase, frequency etc. For example, on Figure 4.2, we have randomized the frequency. Given the randomness of this signal, it is difficult for the attacker to flip symbol “1” to “0” as it would need to predict the shape of the random signal in order to cancel it.

However, generating arbitrarily random signals using off-the-shelf wireless devices, is challenging and, with most devices, not feasible. This is mainly due to the implemented signal modulation schemes that does require the bits to be encoded in a predefined fashion (e.g., in the case of FSK modulation, symbols “1” are transmitted as a sinusoid waveform at one frequency, and symbols “0” is transmitted as the same waveform but at a different frequency). We therefore propose a simple, yet effective solution to randomize the transmission of symbol “1”, which is compatible with the underlying modulation schemes. For this, we introduce an additional step of encoding called *signal spreading*. This is shown on Figure 4.3. An  $I$ -coded message  $c$  is spread such that symbols “1” are converted into random sequences of  $k$  chips each; symbols “0” are converted into *null* symbols. On the channel, chips “1” and “0” are transmitted using the modulation scheme available to the sender (in our example we use FSK modulation), whereas the *null* symbol is transmitted as the absence of signal.

Assuming that the designated receiver can demodulate the signal at approximately the same speed as the attacker, the ability of the attacker to flip symbol “1” to “0” essentially depends on his ability to guess one of the chip sequences. If the attacker fails to guess the entire sequence, the receiver will still (correctly) decode this signal into symbol “1”. The probability that the attacker guesses the chip sequence of a specific bit is  $2^{-k}$ . For the fixed codeword  $c$ , the attacker’s probability

to flip one of the symbols “1” is therefore

$$1 - \left(1 - 2^{-k}\right)^n \approx 1 - e^{-n/2^k},$$

where  $n$  is the number of symbols “1” in  $c$  and the approximation is valid for small  $2^{-k}$ . For example, if  $k = 48$  and  $n = 80$ , this probability is  $2^{-40}$ . Obviously, by increasing  $k$ , this probability can be made arbitrarily small.

In Section 4.6, we treat this issue of ensuring that the attacker cannot cancel the signal in greater detail.

#### 4.3.4 Synchronization and Complementary Encoding

Thus far, we have assumed that the sender and the receiver are synchronized with respect to the beginning and the end of the transmission of the given codeword  $c$ . In this section, we show how this can be achieved. Let us start with a simple example.

**Example 3 (Straightforward Synchronization)** Assume that Alice meets Bob and wants to send a message  $m$  to him, using the  $I$ -codes approach. In this scenario, a simple synchronization scheme would consist of using codewords of a fixed length that is publicly known, and letting Alice check if Bob is listening on the correct channel, before she starts transmitting the message. In order to let Bob’s device know when it should start demodulating the message transmitted, we can use the convention that every  $I$ -code codeword is prefixed with symbol “1”. When Alice finishes with the transmission, she informs Bob who, in turn, “notifies” his device (e.g., by a push on a button). In this way, Bob informs his device that it may begin to demodulate the received message. The important point is that the Bob’s device should take into account all the symbols it received between the time instant at which the first symbol “1” has arrived and the time instant at which Bob has notified his device (i.e., the push on the button).

As far the synchronization at the physical layer is concerned, by appropriately setting  $T_s$ , we can easily ensure that the transmitter and the receiver remain synchronized throughout the transmission. In Section 4.4, we report on our experience with a concrete real-life implementation.

Clearly, the approach to the synchronization from the previous example is not very flexible. We next describe a more flexible approach. Let us assume that the sender wants to transmit the following codeword  $c = 1010011001$  (which corresponds to the message  $s = 11010$  under the complementary encoding rule). The sender simply keeps emitting (using the on-off keying, Figure 4.2) the following repetitive sequence

$$\dots \text{delimiter} \overbrace{1010011001}^c \text{delimiter} \overbrace{1010011001}^c \text{delimiter} \dots \quad (4.1)$$

Here, the “delimiter” represents a specially constructed bit string such that any *successfully demodulated codeword*<sup>2</sup> received between any two consecutive “delimiters” is authentic (i.e., corresponds to 1010011001 in our example). We will show shortly how to construct such a delimiter for the complementary encoding rule.

The receiver first has to make sure that the peer sender is active (transmitting the above repetitive sequence). Then it decodes a codeword received between any two consecutive “delimiters”.

---

<sup>2</sup>In our example, by “successfully demodulated codeword” we mean the codeword for which the transformation ( $10 \rightarrow 1, 01 \rightarrow 0$ ) exists.



If the codeword can be converted back to a message using the inverse of the complementary encoding rule (i.e.,  $(10 \rightarrow 1, 01 \rightarrow 0)$ ), the receiver accepts this message as being authentic. At this stage, the peer sender can stop transmitting the above repeated sequence. A nice property of this approach is that the receiver does not have to know the length of the codeword being transmitted in advance.

We next define more formally the notion of the “delimiter”. Then we construct the delimiter for the complementary encoding rule.

**Definition 10** *For the fixed set of codewords  $\mathcal{C}$ , we define an incongruous delimiter (shortly,  $i$ -delimiter) to be a finite minimum-length string of bits that satisfies the following conditions:*

1. *No substring (of consecutive bits) of any codeword  $c \in \mathcal{C}$  can be converted into the  $i$ -delimiter, without flipping at least one bit “1” of  $c$  to bit “0”;*
2. *The  $i$ -delimiter cannot be converted into a substring (of consecutive bits) of any  $c \in \mathcal{C}$ , without flipping at least one bit “1” of the  $i$ -delimiter to bit “0”;*
3. *Any valid codeword (i.e., any  $c \in \mathcal{C}$ ) received between two consecutive  $i$ -delimiters is authentic.*

**Example 4** Consider the set  $\mathcal{C}$  such that  $c = 10100110 \in \mathcal{C}$ . Consider also the following candidate for the  $i$ -delimiter:  $x = 11011$ . We will show that bit-string  $x$  does not satisfy Definition 10 and therefore is not an  $i$ -delimiter for the set  $\mathcal{C}$ . This is easily seen by observing that  $10100110 \rightarrow 10110110$ , i.e., it is sufficient to flip only the fourth bit of  $c$  so that  $x$  emerges as the substring of  $c$ . Therefore, the first condition of Definition 10 is not met.

Assuming that an adversary cannot flip bit “1” into bit “0”, we have the following result.

**Theorem 13** *Consider the set of codewords  $\mathcal{C}$  obtained by applying the complementary encoding rule ( $1 \rightarrow 10, 0 \rightarrow 01$ ) to the set of source states (messages)  $\mathcal{S} = \{0, 1, 00, 01, \dots, \overbrace{11 \dots 1}^k\}$ , for arbitrary  $k < \infty$ . A string 111000 is an  $i$ -delimiter for the set  $\mathcal{C}$ .*

*Proof:* By mere inspection of all the strings of a length smaller than 6 bits, it easily follows that no such string satisfies Definition 10.

Consider now the string 111000. Observe that for every codeword  $c \in \mathcal{C}$  the number of consecutive bits 0 and the number of consecutive bits 1 is at most two. Therefore, (i) 111000 cannot be converted into any codeword  $c \in \mathcal{C}$  without flipping at least one of the leading bits “1” in 111000 to bit “0”, and (ii) no substring of any codeword  $c \in \mathcal{C}$  can be converted into 111000, without flipping at least one bit “1” of  $c$  to bit “0”. Thus, the string 111000 satisfies the first two conditions in Definition 10.

We next show that it satisfies the third condition as well. We observe that it is sufficient to focus on a codeword between two consecutive strings 111000, since three consecutive bits “1” never appear in any valid codeword from  $\mathcal{C}$  and the adversary cannot flip a bit “1”. Let us consider the following sequence of bits for any  $k$ -bit codeword ( $k$  being even)  $c = (c_1 c_2 \dots c_{k-1} c_k) \in \mathcal{C}$

$$\dots 111000 c_1 c_2 \dots c_{k-1} c_k 111000 \dots \quad (4.2)$$

We first show that the adversary cannot accomplish that the string 111000 emerges in any (other) part of the sequence (4.2) and that at the same time any resulting codeword  $\hat{c}$  is valid. As the result the only hope for the adversary is to leave the original delimiters 111000 intact and try

to transform the original codeword  $c$  into a different codeword  $\hat{c}$  of the same length. Since  $c$  is an  $I$ -code codeword, the adversary would have to flip at least one bit “1” of  $c$  into a bit “0”. However, by assumption he cannot accomplish this.

We now prove that the adversary cannot achieve that the 111000 emerges in any (other) part of the sequence (4.2) and that at the same time any resulting codeword  $\hat{c}$  is valid. For this, let us consider all possible 6-bit substrings (of consecutive bits) in the sequence (4.2). These can be captured by one of the eleven cases given below:

1. ... 1 11000 $c_1$   $c_2 \dots c_{k-1} c_k$  111000 ...
2. ... 11 1000 $c_1 c_2$   $c_3 \dots c_{k-1} c_k$  111000 ...
3. ... 111 000 $c_1 c_2 c_3$   $c_4 \dots c_{k-1} c_k$  111000 ...
4. ... 1110 00 $c_1 c_2 c_3 c_4$   $c_5 \dots c_{k-1} c_k$  111000 ...
5. ... 11100 0 $c_1 c_2 c_3 c_4 c_5$   $c_6 \dots c_{k-1} c_k$  111000 ...
6. ... 111000 $c_1 c_2 \dots c_{i-4}$   $c_{i-3} c_{i-2} c_{i-1} c_i c_{i+1} c_{i+2}$   $c_{i+3} \dots c_{k-1} c_k$  111000 ...
7. ... 111000 $c_1 c_2 \dots c_{k-5}$   $c_{k-4} c_{k-3} c_{k-2} c_{k-1} c_k 1$  111000 ...
8. ... 111000 $c_1 c_2 \dots c_{k-4}$   $c_{k-3} c_{k-2} c_{k-1} c_k 11$  1000 ...
9. ... 111000 $c_1 c_2 \dots c_{k-3}$   $c_{k-2} c_{k-1} c_k 111$  000 ...
10. ... 111000 $c_1 c_2 \dots c_{k-2}$   $c_{k-1} c_k 1110$  00 ...
11. ... 111000 $c_1 c_2 \dots c_{k-1}$   $c_k 11100$  0 ...

*Case 2 – Case 5.* The strings  $(1000c_1c_2)$ ,  $(000c_1c_2c_3)$ ,  $(00c_1c_2c_3c_4)$  and  $(0c_1c_2c_3c_4c_5)$  cannot be transformed into the string 111000 without flipping at least one bit “1”, since  $c_1 \oplus c_2 = 1$  and  $c_3 \oplus c_4 = 1$  (by the complementary encoding).

*Case 6.* We showed at the beginning of the proof that the string 111000 satisfies the condition one in Definition 10. So no string  $(c_{i-3}c_{i-2}c_{i-1}c_i c_{i+1}c_{i+2})$ ,  $i \in [4, 5, \dots, k-2]$ , can be transformed into the string 111000 without flipping at least one bit “1”.

*Case 7 – Case 11.* The strings  $(c_{k-4}c_{k-3}c_{k-2}c_{k-1}c_k 1)$ ,  $(c_{k-3}c_{k-2}c_{k-1}c_k 11)$ ,  $(c_{k-2}c_{k-1}c_k 111)$ ,  $(c_{k-1}c_k 1110)$  and  $(c_k 11100)$  cannot be converted into the string 111000 without flipping at least one bit “1”, since they all contain at least one bit “1” among the last three digits.

*Case 1.* The string  $(11000c_1)$  can be transformed into the string 111000 by flipping the third bit to “1”, conditioned on  $c_1 = 0$ . In this case, the bit  $c_2 = 1$  becomes the first bit of the new codeword  $\hat{c}$  (not necessarily valid). From *Case 2 – Case 11* above we know that the ending of the codeword  $\hat{c}$  must be denoted either by the original delimiter 111000 or by the delimiter obtained by joining the first bit “1” of the original delimiter to the new codeword  $\hat{c}$ . In the first case, the length of the resulting codeword  $\hat{c}$  is  $k-1$  (an odd number) and so  $\hat{c}$  cannot be a valid codeword. In the second case, one bit “1” is added to the sequence that already has a deficit of bits “0” (i.e., the bit  $c_1 = 0$  is not a part of  $\hat{c}$ ) and so the resulting codeword  $\hat{c}$  cannot be not valid.

We conclude the proof by observing that the string 111000 is the shortest string (i.e., 6 bits long) that satisfies all the conditions in Definition 10.  $\square$

**Remark 2** *It is interesting to observe that for the complementary encoding rule and the delimiter 111000, the first two conditions from Definition 10 imply the third one (they are sufficient). If this holds in general (for any  $I$ -code and an  $i$ -delimiter) is an interesting open problem.*

Referring back to the example (4.1), the sender can preserve the integrity of message 11010 (i.e., the codeword  $c = 1010011001$ ) by simply emitting (using the on-off keying) the following repetitive sequence

$$\dots \underbrace{111000}_{i\text{-delimiter}} \overbrace{1010011001}^c \underbrace{111000}_{i\text{-delimiter}} \overbrace{1010011001}^c \underbrace{111000}_{i\text{-delimiter}} \dots$$

The receiver decodes a codeword received between any two consecutive  $i$ -delimiters (after having verified that the peer sender is active). According to Theorem 13, any successfully demodulated codeword between two  $i$ -delimiters must have been emitted by the peer sender (the codeword is authentic). At this stage, the peer sender can stop transmitting the above repeated sequence. The important implication of the synchronization based on  $i$ -delimiters is that the receiver does not have to know in advance the length of the message to be transmitted by the sender.

In the following sections, we report on our experience with the real-life implementation of  $I$ -codes and we describe the usage of  $I$ -codes for broadcast authentication and key agreement.

## 4.4 Implementation

We implemented  $I$ -codes (with spreading) on the Mica2 sensor networking platform [2]. This platform consists of a processor and a CC1000 radio. CC1000 is a single-chip RF transceiver, has a programmable frequency (300-1000 MHz) and uses FSK modulation spectrum shaping. It has programmable output power, (-20 to 10 dBm) and a high receiver sensitivity (-110 dBm).

In our  $I$ -code implementation, we use pairs of sensors running the SOS operating system [46]. Each original message  $m$  is first  $I$ -coded such that each “1” is transformed into a “10” and “0” into a “01”. An  $I$ -coded message is then transmitted such that each “1” is transmitted as an SOS packet containing a random payload of length  $k$  (payload is chosen randomly for each packet) and each “0” is transmitted as an absence of signal of duration  $T_s$  (in our implementation the number of chips per symbol “1” is  $k = 48$  bits and  $T_s = 10$  ms – Figure 4.3). Each packet consists of a preamble and of a payload. The preamble is 12 bytes long and with the payload makes a total of 18 bytes per packet.

The decoding process at the receiver is implemented as follows. A “silence period” on the channel of the duration of 10 ms is interpreted as a “0”, whereas a presence of a packet is interpreted as “1”. Here, the “silence on the channel” is defined as a period during which the received signal strength on the receiver remains below a preset RSSI level. If the signal level remains above the preset RSSI level, but the received information cannot be interpreted as a packet, the signal is interpreted as “1”.

We experimented with this implementation of  $I$ -codes, by sending 8 to 512 bits long messages (pre-coded messages from 16 to 1024 bits). To transmit an  $\ell$ -bit long message using  $I$ -codes we actually transmit  $\ell$  “0”s (10 ms of the absence of signal) and  $\ell$  random packets (each 18 bytes long). We measured the message transmission success  $\rho_t$  as a ratio between the number of correctly transmitted messages and the total number of attempts. Here, we consider that a message is correctly transmitted if the message originating from the sender is the same one received by the receiver. For each different message size, we perform 20 experiments as follows. We first generate 100 random messages of the given size. Next, we transmit these 100 messages and count the number of messages that have been successfully received. From this we calculate the success ratio  $\rho_t$ . Finally, we average the results obtained from 20 experiments and present them with 95% confidence interval.

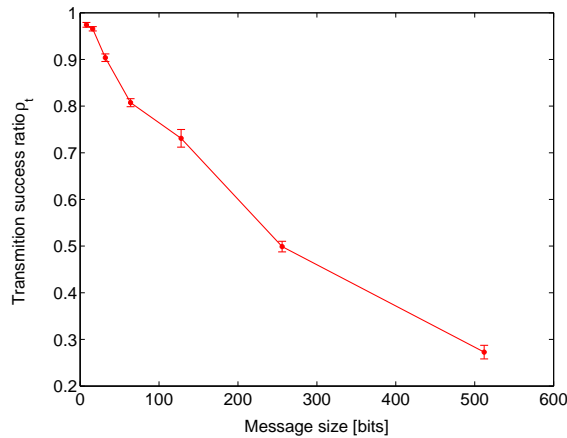


Figure 4.4: Robustness of  $I$ -codes. The figure shows the message transmission success ratio  $\rho_t$  as a function of the size of transmitted messages. The results are obtained through measurements on Mica2 sensor nodes.

The results of our measurements are shown on Figure 4.4. Quite expectedly, from Figure 4.4 we can observe that the transmission success ratio decreases quickly as the message size increases. These results further show that  $I$ -codes are best suited for reasonably short messages. For longer messages, we would need to transmit them multiple times in order for one of the messages to be transmitted correctly. For this purpose, we rely on the  $i$ -delimiters introduced in Section 4.3.4. From our measurement results we further observed that no messages were altered on the channel such that they appear to the receiver as correct  $I$ -coded messages, but they are different from the messages sent by the sender. Moreover, with our implementation, no bit “1” sent by the transmitter was interpreted as a bit “0” on the receiver’s side. This is important as it shows that the integrity of the messages transmitted with  $I$ -codes is preserved in our implementation.

From these measurements we conclude that  $I$ -codes provide sufficient robustness for the transfer of short messages (e.g., public keys, public parameters, message digests, etc). For example, a 160 bit message (a typical size of the message digest) has a 70% chance of being transmitted correctly, meaning that transmitting it correctly with a 0.999 probability takes approximately 6 successive transmissions; on average it will take  $1/0.7 \approx 2$  retransmissions. These numbers can, however, vary depending on the channel conditions (the level of interference on the channel can be also estimated by the sender and taken into account in estimating the number of transmissions).

With the Mica2 communication speed of 19.2 Kbps, each packet (representing a “1”) is transmitted in 7.5 ms. This means that each bit of the original message gets transmitted in 17.5 ms (single “0” and a single “1”) which means that the communication speed of transmitting the original message with  $I$ -codes is 57 bps. Although  $I$ -codes reduce the speed of communication, this speed is sufficient to enable the integrity-preserving transmission of a message digest (the size of which typically is 160 bits), which then guarantees the integrity-preserving transmission of the entire message.

Furthermore, in some scenarios, only the integrity of a public key needs to be preserved, whereas protecting the rest of the communication can be enabled using the previously transmitted public key.

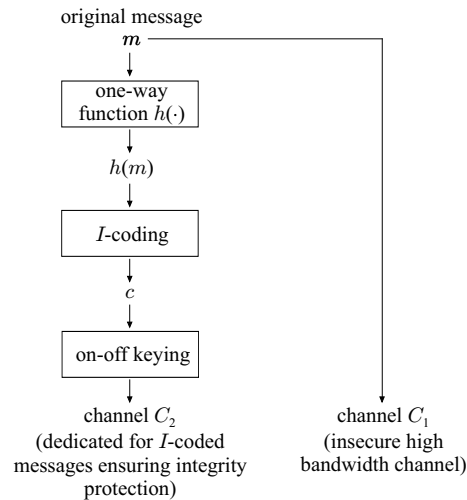


Figure 4.5: Typical usage of  $I$ -codes for integrity protection. Original message is transmitted over an insecure high-bandwidth channel  $C_1$ , whereas the integrity protection is enabled with  $I$ -codes on a different channel  $C_2$ .

## 4.5 Authentication Through Presence

Using  $I$ -codes, we develop a novel concept called *authentication through presence*, which enables (broadcast) message authentication based solely on the awareness of the presence in a power range of an entity. We first introduce this concept and then we describe its use in two application scenarios: broadcast authentication and key establishment.

We describe our concept through an example involving two parties: the sender  $A$  and the receiver  $B$ . Note that the sender and the receiver do not share *any* authentication material. The main idea of our approach is shown on Figure 4.5. The message  $m$ , whose integrity needs to be protected, is sent over a channel  $C_1$  which does not protect its integrity and over which its authenticity cannot be verified. This channel can be realized as *any* communication channel. The message digest  $h(m)$  (e.g., the message hash) is sent over a separate communication channel  $C_2$ , dedicated for integrity protection (we have shown through our implementation in Section 4.4 that this dedicated channel can be realized using existing communication channels). Thus, if  $A$  wants to send a message to  $B$ , she will use the protocol shown on Figure 4.6.

In this protocol,  $h(\cdot)$  represents a one-way function used to protect the integrity of the transmitted message. This function can be implemented as a simple hash.  $I\text{-code}(h(m))$  represents the  $I$ -coded message digest  $h(m)$ . The sequences preceding and following after  $I\text{-code}(h(m))$  are  $i$ -delimiters (Section 4.3.4), which ensure that the receiver knows the beginning and the end of the  $I$ -coded message.

In this protocol, the integrity and the authenticity of the message  $m$  is verified through the verification of the authenticity and integrity of its digest  $h(m)$ . The authenticity and the integrity of  $h(m)$  is guaranteed with  $I$ -codes if and only if the following conditions are met: (i) the receiver  $B$  knows that it is in the power range of the sender  $A$ , (ii) the receiver  $B$  knows that  $A$  has started transmitting on the integrity channel ( $C_2$ ). The first condition is *the condition of presence*, which ensures that the receiver is receiving signals from the sender. The second condition is the *condition of synchronization*, which ensures that the receiver knows at what time the transmission of data

$$\begin{array}{l}
 A \rightarrow B \text{ (on } C_1 \text{)} : m \\
 A \rightarrow B \text{ (on } C_2 \text{)} : \dots \underbrace{111000}_{i\text{-delimiter}} I\text{-code}(h(m)) \underbrace{111000}_{i\text{-delimiter}} \dots \\
 \\
 B : \text{ Verify the integrity and the authenticity} \\
 \quad \text{of } h(m) \text{ using } I\text{-codes.} \\
 \quad \text{Verify the integrity and the authenticity} \\
 \quad \text{of } m \text{ using } h(m).
 \end{array}$$

Figure 4.6: A protocol enabling the *authentication through presence* property;  $h(\cdot)$  represents a one-way function.

takes place. If the receiver wrongly believes that the transmitter is transmitting, or if it wrongly believes to be in the power range of the sender, a (malicious) entity can insert false data on the channel and these data will be accepted as valid by the receiver. This follows from the properties of  $I$ -codes, which assume the presence of the signal from the legitimate sender on the channel.

In the following two sections, we show in which scenarios the conditions of presence and synchronization are fulfilled and in which, therefore,  $I$ -codes can be used for authentication and integrity protection.

#### 4.5.1 Access Point Authentication

In this section, we show that authentication through presence can be a useful tool for the broadcast authentication of messages from fixed access points (AP).

Our scenario is depicted on Figure 4.7. Here,  $I$ -codes are used by the AP to advertise its public key. This key can be later used to provide authentication and integrity protection of all messages generated by the AP.

This enables any user that comes into the range of the AP to know that the advertised public key of this access point is authentic and belongs to the access point in whose range they lie. If the user trusts the environment in which the access point is placed (a bank or an office), it will trust all information coming from that access point and will use the public key of the AP to establish a secure connection to the station. Here, it is important that the user knows that the environment in which she is placed is covered by at least one legitimate AP. If this condition is fulfilled, it is of little importance if there are any rogue APs present in this space, as long as the legitimate APs are active.

We assume that the sender (AP) is static. The (conservative) reach of its transmission is known to the receivers. The receivers therefore know before they start receiving the data if they are in the sender's power range or not; this knowledge is publicly available information. The receivers also know the integrity channel used by the AP to emit its public key. In the case of, for example, IEEE 802.11a, one of the 12 orthogonal channels can be allocated for this purpose.

The AP continuously sends its key on the integrity channel ( $C_2$  on Figure 4.5). When it is not advertising its public key, the AP jams the integrity channel to prevent any fake public keys being transmitted over the same channel. As the AP is continuously active, there is no need for synchronization with the receivers; the receivers will start receiving the data when they come into AP's power range. This power range can be estimated by the receiver (a room where the AP is

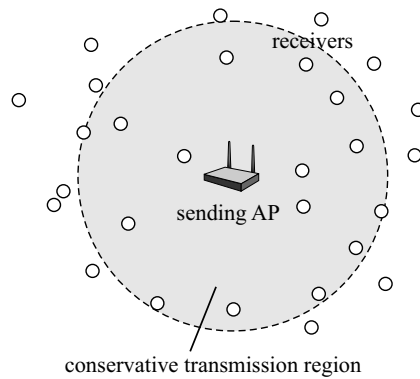


Figure 4.7: Broadcast integrity and authentication with an access point. By the “conservative transmission region” we mean the region where the received power of a signal transmitted by the AP exceeds some predefined threshold level (which is a security parameter in our case).

placed), or can even be marked. Furthermore, to avoid attacks during the time when the AP fails, its status (activity) can be signalled to the receivers through some visual channel (e.g. a blinking LED).

#### 4.5.2 Key Establishment over Insecure Channels

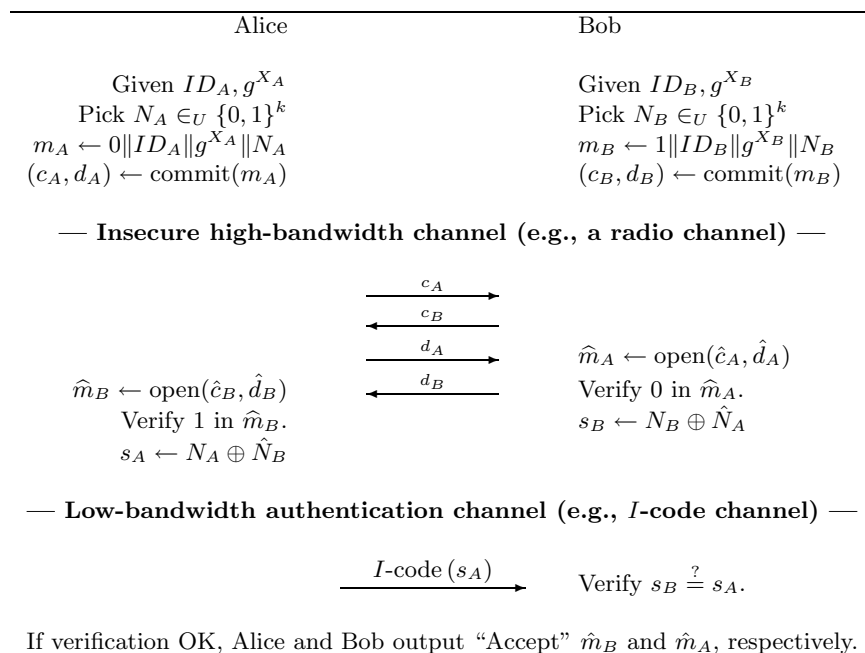
In this section we show how authentication through presence can be used for key establishment over an (insecure) radio link in peer-to-peer networks. Our key establishment protocol is based on the Diffie-Hellman key agreement with String Comparison (DH-SC) shown on Figure 3.6 in Chapter 3. The protocol that we propose in this section (Figure 4.8) is essentially the same protocol as the DH-SC protocol. The only difference is that instead of having the users compare the short authentication strings  $s_A$  and  $s_B$  via face-to-face voice or visual communication, the authentication string  $s_A$  (or  $s_B$ ) is communicated using  $I$ -codes (see Figure 4.8). We call this protocol the DH-IC protocol.

Clearly, the security analysis of the DH-SC in Section 3.4.2, extends to the DH-IC protocol. By combining Proposition 2 (Chapter 3) and the analysis of the security characteristics of  $I$ -codes from Section 4.6, we have the following result; again, we denote with  $\gamma$  the maximum number of sessions (successful or abortive) of the DH-IC protocol that any party can participate in.

**Theorem 14** *The probability that an attacker succeeds against a targeted user of the DH-IC protocol (with the  $\text{commit}(\cdot)$  being perfectly hiding and computational binding) is bounded by  $\gamma 2^{-k} + \varepsilon(\text{par})$ , where  $\varepsilon(\text{par})$  is negligible in the security parameter( $s$ )  $\text{par}$  of the used commitment scheme.*

Here, we assume that prior to the protocol execution, the entities know the system parameters and are aware of each others’ presence in the communication range. Therefore, the following condition must be met: the sender has to make sure that the receiver is turned on and is listening on the (correct) channel during the sender’s transmission. This can be easily enforced if two users approach each other to establish a common secret key.

Let us give an example of possible values for the above parameters. Assume that any party can participate in at most  $\gamma = 2^{30}$  sessions (successful or abortive) in its lifetime. Then, by choosing  $k = 60$  we obtain that the highest probability of success by the attacker (having seen a huge number  $\gamma = 2^{30}$  of protocol runs) is approximately  $\gamma 2^{-k} = 2^{-30}$ . Note that  $k$  also represents the length of the

Figure 4.8: Diffie-Hellman key agreement protocol based on  $I$ -codes (DH-IC)

verification string  $s_A$  (and  $s_B$ ) to be communicated through  $I$ -codes. From Figure 4.4, we can see that with  $I$ -codes, in normal circumstances, it will take on average around  $(1/0.7) < 2$  repetitions of the message of length  $k = 60$  bits (i.e., 120 bits long codeword with the complementary encoding), before it is successfully received by the given receiver. This is rather negligible cost, given that all the messages are transmitted over a radio link.

Therefore, with  $I$ -codes, the involvement of the users in the protocol execution is rather minimal.

## 4.6 Security Analysis of $I$ -codes

In this section, we discuss security of  $I$ -codes from the signal cancellation point of view. As we already mentioned in Section 4.3.3, the security of  $I$ -codes depends on the inability of the attacker to flip symbols “1” into “0”, by which she breaks the integrity of the exchanged messages. By a successful attack on  $I$ -codes, we consider that the attacker is able to break the integrity of the transmitted message, meaning that the receiver accepts a message as valid even if it has been modified by the attacker on the channel. Note that we reason about the security of  $I$ -codes within the system and the attacker model described in Section 4.2.

We focus on the security of  $I$ -codes used over the radio communications channel. In order to delete (cancel) a signal  $s(t)$  emitted on a radio channel, the only hope for the adversary is to have its signal  $s'(t)$  arrive at the receiver with the same amplitude as  $s(t)$  but opposite in phase, that is,  $s'(t) = -s(t)$ . There are two main factors that make it hard for the attacker to cancel the signal at the receiver: (1) the unpredictability of the channel conditions (2) the unpredictability of the signal generated by the sender. In order to cancel the signal at the receiver, the attacker needs to estimate the channel conditions (to know how the channel will shape the original signal), and predict the shape of the signal generated at the sender (to know which form to generate to cancel



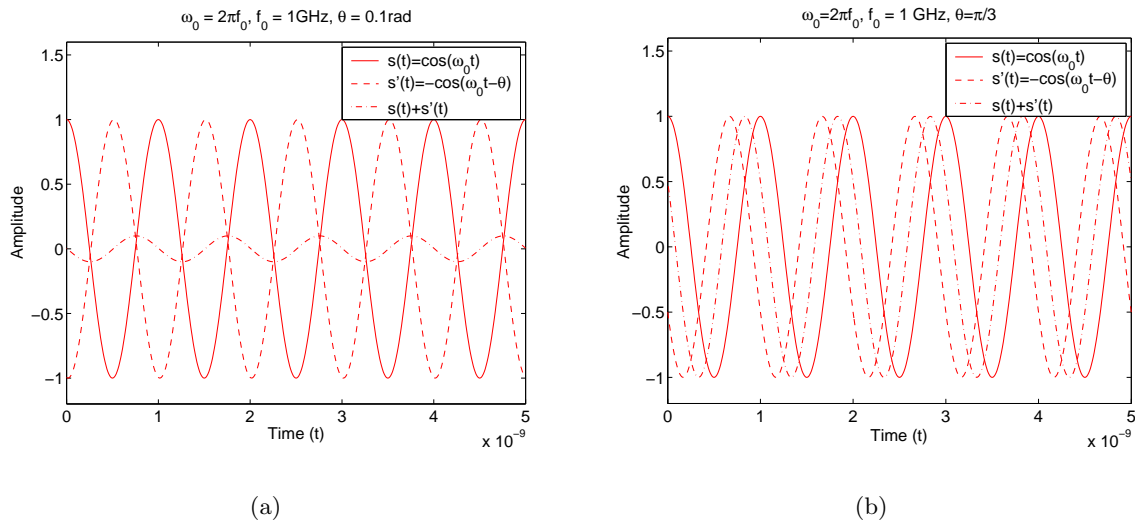


Figure 4.9: Superposition of signals  $s(t) = \cos(\omega_0 t)$  and  $s'(t) = -\cos(\omega_0 t - \theta)$ : (a)  $\theta = 0.1$  rad; (b)  $\theta = \frac{\pi}{3}$  rad.

the signal). Channel conditions are highly influenced by the environment and in high-frequency communication systems (e.g., 2.4 GHz), it is nearly impossible for the attacker to predict them due to the unpredictable amplitudes and phases, the multipath fading effects, etc.

In this section, we analyze how channel and signal unpredictability affect the attacker's ability to cancel-out the signal on the channel. We show that the odds of the adversary to cancel the signal  $s(t)$  are indeed negligible.

#### 4.6.1 Anti-Blocking Property of the Radio Channel

We first start by showing how channel conditions affect the attacker's ability to cancel the radio signal.

Let us assume that the sender emits cosine signal  $s(t)$  with unit amplitude and frequency  $f_0$ , i.e.,  $s(t) = \cos(\omega_0 t)$ , where  $\omega_0 = 2\pi f_0$ . We assume that the adversary somehow knows the exact value of the amplitude of the signal received at the receiver. Furthermore, we assume that there are no multipath fading effects and that the adversary knows  $s(t)$ . Note that with these assumptions, we only make the task of the adversary much easier. In reality, multipath effects and interferences from other transmitters can easily make the channel sufficiently random to forbid the attacker to even estimate the state of the signal at the receiver  $r(t)$ .

Let us define  $r(t) \equiv \cos(\omega_0 t) - \cos(\omega_0 t - \theta)$ , where  $\theta \in [0, 2\pi)$ . Here,  $r(t)$  can be thought of as the signal obtained as the superposition of the adversary's annihilating signal  $s'(t) = -\cos(\omega_0 t - \theta)$  and  $s(t)$ ;  $\theta$  accounts for the potential *phase shift*. On Figure 4.9 we plot signal  $r(t) = s(t) + s'(t)$  for two different phase shifts:  $\theta = 0.1$  rad and  $\theta = \pi/3$  rad, respectively. The energy  $E_r$  of the signal

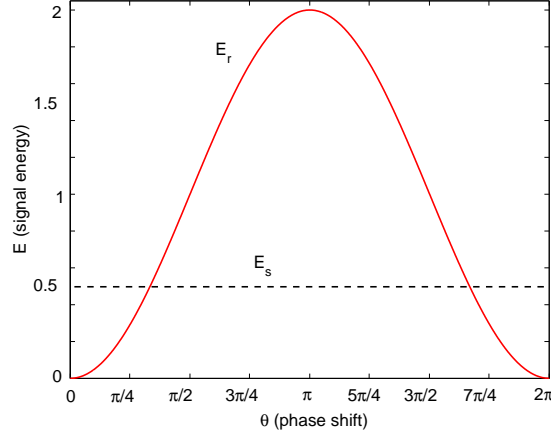


Figure 4.10: The energy of the signal  $r(t) \equiv \cos(\omega_0 t) - \cos(\omega_0 t - \theta)$  and the signal  $s(t) = \cos(\omega_0 t)$  normalized with respect to  $T_s$  (the average power).

$r(t)$ , with duration  $T_s$ , can be calculated as follows [88]:

$$\begin{aligned}
 E_r &= \int_0^{T_s} r^2(t) dt \\
 &= \frac{1}{\omega_0} \sin^2\left(\frac{\theta}{2}\right) (2\omega_0 T_s - \sin(\theta) + \sin(\theta - 2\omega_0)) \\
 &\stackrel{(1)}{\approx} 2T_s \sin^2\left(\frac{\theta}{2}\right),
 \end{aligned} \tag{4.3}$$

where the approximation (1) is valid for high frequencies  $f_0$  (e.g.,  $f_0 = 2.4$  GHz), since  $-1 \leq \sin(\cdot) \leq 1$  implies  $\sin(\cdot)/\omega_0 = \sin(\cdot)/(2\pi f_0) \rightarrow 0$ .

We plot the expression (4.3) on Figure 4.10; note that we normalize the energy with respect to  $T_s$  (therefore obtaining the average power of the signal). On the same figure, we also plot the energy of the unobstructed signal  $s(t) = \cos(\omega_0 t)$ , i.e.,  $E_s = \int_0^{T_s} \cos^2(\omega_0 t) dt = T_s/2$ . A striking result on this figure is that for most values of  $\theta$  the adversary actually contributes to the energy of the original signal  $s(t)$ . In order to at least attenuate  $s(t)$ , the adversary has to ensure that  $\theta \in (-\theta_0, \theta_0)$ , where  $\theta_0$  is calculated as follows:

$$\frac{E_r}{E_s} = 4 \sin^2\left(\frac{\theta}{2}\right) < 1 \Rightarrow \sin\left(\frac{\theta}{2}\right) < \pm \frac{1}{2}, \tag{4.4}$$

and therefore,  $\theta_0 = 2 \arcsin\left(\frac{1}{2}\right) = \frac{\pi}{3}$ . Therefore, the attacker attenuates<sup>3</sup>  $s(t)$  for  $\theta \in [0, \frac{\pi}{3}] \cup (\frac{5\pi}{3}, 2\pi]$  (see Figure 4.10); note that this interval represents  $1/3$  ( $\approx 33\%$ ) of all the possible phase shifts.

We now show how demanding it is for the attacker to keep the phase shift  $\theta$  within the given bounds. We know that  $\theta = \omega_0 \Delta t$ , for a *time shift (delay)*  $\Delta t$ . In time  $\Delta t$ , the electromagnetic wave can travel the distance  $\Delta d = \Delta t \cdot c$ , where  $c$  is the propagation speed of the wave. We call  $\Delta d$  the *distance shift*. Combining these expressions we have:

$$\theta = \frac{2\pi f_0}{c} \Delta d. \tag{4.5}$$

<sup>3</sup>Not necessarily causing sufficient signal attenuation.

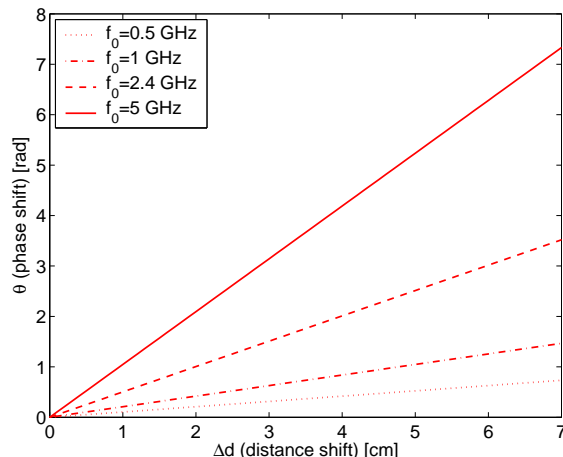


Figure 4.11: The phase shift  $\theta$  as a function of the distance shift  $\Delta d$  for different frequencies  $f_0$ .

On Figure 4.11 we plot expression (4.5) for different frequencies  $f_0$ . We can see that the higher the frequency of the signal is, the higher the effect of the fixed distance shift  $\Delta d$  on the phase shift  $\theta$  is. More importantly, for  $f_0 = 5$  GHz (IEEE 802.11a), a  $\Delta d$  as small as 1 cm results in phase shift of  $\frac{\pi}{3}$ . As we discussed above, the adversary has to ensure that  $\theta \in [0, \frac{\pi}{3}] \cup (\frac{5\pi}{3}, 2\pi]$ , in order to at least attenuate the signal  $s(t)$ . A more reasonable goal for the adversary would be to reduce the energy of the signal  $s(t)$  for say 50%, which requires, for  $f_0 = 5$  GHz,  $\theta \in [0, 0.7227] \cup (5.5605, 2\pi]$ . This phase shift corresponds to  $\Delta d \approx 7$  mm. Therefore, for high frequencies, the adversary has to estimate the distances between himself and both the sender and the receiver with a very high accuracy. Otherwise, he cannot hope to have the phase shift fall within the desired interval.

If the distance between the sender and the receiver continuously changes (in a fashion unpredictable to the attacker), the uncertainty of the adversary is further increased (note that this can be a very limited motion, in the order of  $\Delta d$ ). Therefore, indeed, *mobility helps security* [26]. Another source of the uncertainty for the adversary is the time delay  $\Delta t = \Delta d/c$ . For example, a distance shift  $\Delta d = 7$  mm is equivalent to a delay of  $\Delta t \approx 23$  ps. Therefore, the adversary has to operate with an extremely high time accuracy, otherwise he cannot keep  $\theta$  within the desired bounds, at least not deterministically.

Finally, if we assume that the receiver is equipped with two (or more) mutually separated antennas (as in *multiple antenna systems* [88]), then a signal from some transmitter will most likely arrive at the antennas with different phases. Moreover, this shift between the phases of the signals received by will depend on the distances between the antennas, as well as the relative position of the attacker with respect to the antennas. As we have already seen above, at very high frequencies, even a very small distance shift will cause a significant phase shift. Any uncertainty in the distance shift (e.g., due to distance estimation errors, uncertainty regarding the positions of the antennas, etc.) implies uncertainty in the phase shift. We therefore conclude that it is reasonable to model phase shift  $\theta$  by a random variable with appropriate distribution.

#### 4.6.2 Randomization at the Sender: the Impact of Spreading

We have already seen in Figure 4.10 that for 1/3 of the possible phase shifts, the adversary actually attenuates the sender's signal. Therefore, when using only a single waveform (e.g.,  $\cos(\omega_0 t)$ ) during

the whole period  $T_s$ , the adversary may have a non-negligible probability to attenuate the desired signal. For example, assuming  $\theta$  is a sample of a random variable  $\Theta$  with uniform distribution on  $[0, 2\pi)$ , the adversary attenuates the signal in the single time interval  $T_s$  with probability  $1/3$ . We now apply a solution similar to spreading, already described in Section 4.3.3.

The idea is to split the time interval  $T_s$  into  $K$  smaller and equal time slots  $T_m$  when the symbol “1” is to be sent. Then, for each *mini-slot*  $T_m$ , the sender generates a signal with the phase chosen uniformly at random from  $[0, 2\pi)$  and emits these  $K$  signals on the channel during the time  $T_s$ . For example, these  $K$  signals can be described by the following random process  $S(t) = \cos(\omega_0 t + \Phi)$ , where  $\Phi$  is a random variable with uniform distribution on  $[0, 2\pi)$ .

From the discussion in the previous section, it is reasonable to model the phase shift as a random variable  $\Theta$ . Let us assume  $\Theta$  to be uniformly distributed on  $[0, 2\pi)$ ; later in this section, we also consider Gaussian distribution. Let  $p_\alpha$  be the probability that the adversary attenuates the signal emitted in a given mini-time slot for at least  $(1 - \alpha) \times 100\%$ , that is,  $E_r/E_s \leq \alpha$ , where  $\alpha \in [0, 1]$ . We say that any such mini-slot signal is  $\alpha$ -attenuated<sup>4</sup>. For  $\Theta$  uniform random variable, i.e.  $f_\Theta(\theta) = \frac{1}{2\pi}$ , we have

$$\begin{aligned} p_\alpha &= P \left[ \frac{E_r}{E_s} \leq \alpha \right] \\ &\stackrel{(1)}{=} P \left[ \sin \left( \frac{\theta}{2} \right) \leq \pm \frac{\sqrt{\alpha}}{2} \right] \\ &= P [\theta \in [0, \theta_\alpha] \cup (2\pi - \theta_\alpha, 2\pi)] \\ &\stackrel{(2)}{=} \frac{\theta_\alpha}{\pi}, \end{aligned} \tag{4.6}$$

where  $\theta_\alpha = 2 \arcsin(\sqrt{\alpha}/2)$ , the equality (1) follows from expression (4.4), and the equality (2) follows from the distribution of  $\Theta$ .

We further note that  $\Phi$  and  $\Theta$  are independent random variables; indeed,  $\Theta$  models the inability of the adversary to perfectly estimate the required distances and/or any delay that the adversary introduces. Therefore,  $p_\alpha$  (as given in expression (4.6)), is the same for all the  $K$  mini-slots. Then, for the fixed time interval  $T_s$ , the probability that the number  $K_\alpha$  of  $\alpha$ -attenuated mini-slot signals is exactly  $k \leq K$ , can be calculated from the binomial distribution with parameters  $p = p_\alpha$  and  $q = 1 - p_\alpha$  as follows

$$P[K_\alpha = k] = \binom{K}{k} \frac{1}{\pi^K} \theta_\alpha^k (\pi - \theta_\alpha)^{K-k}, \tag{4.7}$$

where  $\theta_\alpha = 2 \arcsin(\sqrt{\alpha}/2)$ . For the binomial distribution (4.7), we can calculate the expected ratio  $K_\alpha/K$  of the  $\alpha$ -attenuated mini-slots as follows,

$$E \left[ \frac{K_\alpha}{K} \right] = \frac{E[K_\alpha]}{K} = \frac{\theta_\alpha}{\pi} \leq \frac{1}{3}, \tag{4.8}$$

where the last inequality follows from the fact that  $\theta_\alpha \leq \theta_1 = \frac{\pi}{3}$ . Therefore, on average, at most  $1/3$  of the total number of mini-slot signals will be  $\alpha$ -attenuated, i.e.,  $E_r/E_s \leq \alpha$ .

Note, however, that the expected value of the ratio  $K_\alpha/K$  is independent of  $K$ , and therefore it does not give any useful information about the role of  $K$  and what value we should choose for it.

---

<sup>4</sup>Note that even if the adversary does attenuate the energy of the original signal  $s(t)$  by 50%, the average power as measured by the receiver may still be well above the threshold  $P_0$ .

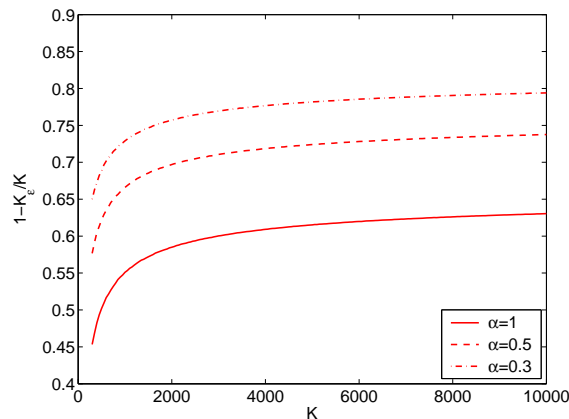


Figure 4.12: The ratio of mini-slot signals that are not  $\alpha$ -attenuated as a function of  $K$ ;  $\epsilon = 10^{-14}$ .

We next study this aspect. Let us denote with  $K_\epsilon$  ( $K_\epsilon \leq K$ ) the *smallest* threshold for which the following holds

$$P[K_\alpha \leq K_\epsilon] \geq 1 - \epsilon, \quad (4.9)$$

where  $\epsilon \in [0, 1]$ . Note that  $P[K_\alpha \leq K_\epsilon] = \sum_{k=0}^{K_\epsilon} P[K_\alpha = k]$ , with  $P[K_\alpha = k]$  given by (4.7). Note further that  $P[K_\alpha \leq K_\epsilon]$  is related to a single time interval  $T_s$  during which the symbol “1” is transmitted. By the independence, the probability  $P^n[K_\alpha \leq K_\epsilon]$  that  $K_\alpha \leq K_\epsilon$  after  $n$  symbol “1” transmissions ( $n$  time intervals  $T_s$ ) satisfies

$$P^n[K_\alpha \leq K_\epsilon] \geq (1 - \epsilon)^n \approx e^{-n\epsilon},$$

where the last approximation is valid for small  $\epsilon$ . For the given  $n$ , by choosing  $\epsilon$  such that  $e^{-n\epsilon}$  is reasonably close to 1, we essentially make  $K_\epsilon$  an “upper bound” on the number of mini-slot signals that are  $\alpha$ -attenuated in any given time slot  $T_s$  (out of the total of  $n$  slots). Likewise,  $(K - K_\epsilon)$  provides a “lower bound” on the number of mini-slot signals that are not  $\alpha$ -attenuated.

On Figure 4.12, we plot the ratio  $(1 - K_\epsilon/K)$  of the mini-slot signals that are not  $\alpha$ -attenuated as a function of  $K$ , for  $\epsilon = 10^{-14}$ . For  $n = 10^{10}$ , we have  $e^{-n\epsilon} \approx 0.9999$ , i.e., even after as many as  $10^{10}$  transmissions of the symbol “1”, the probability that  $K_\alpha \leq K_\epsilon$  is at least 0.9999. If we transmit on average one symbol “1” per second (meaning that we do nothing else but transmit such signals), then it would take around 310 years to see all the  $n$  symbols. In this case, the smallest  $K_\epsilon$  for which the bound (4.9) holds, is a reasonable upper bound on  $K_\alpha$ . Coming back to Figure 4.12, we can see that if  $K$  is set too low, we cannot hope to achieve a very high ratio of non  $\alpha$ -attenuated mini-slot signals for all the  $n$  transmissions of the symbol “1”. Therefore,  $K$  should be chosen based on the expected  $\alpha$  and the desired ratio  $1 - K_\epsilon/K$ .

### 4.6.3 Energy Content of the Emitted Signals

We already argued that it is reasonable to model the phase shift as a random variable  $\Theta \in [0, 2\pi)$ . It is then interesting to calculate the energy of the resulting random signal. Let us define a random process  $R(t) = \cos(\omega_0 t) - \cos(\omega_0 t - \Theta)$ . We will calculate the energy of this process for two different distributions of  $\Theta$ , namely, uniform distribution on  $[0, 2\pi)$  and Gaussian distribution with zero mean and variance  $\sigma_\Theta^2$ .

### Uniform Distribution of $\Theta$

We have  $f_{\Theta}(\theta) = \frac{1}{2\pi}$ ,  $\forall \theta \in [0, 2\pi)$ . The energy content  $\mathcal{E}_R$  of the random process  $R(t)$ , within the time interval  $T$ , is defined as [88]:

$$\mathcal{E}_R = E \left[ \int_0^T R^2(t) dt \right] = \int_0^T E [R^2(t)] dt . \quad (4.10)$$

Now, for  $E [R^2(t)]$  we have:

$$\begin{aligned} E [R^2(t)] &= \int_0^{2\pi} r^2(t) f_{\Theta}(\theta) d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} (\cos(\omega_0 t) - \cos(\omega_0 t - \theta))^2 d\theta \\ &= 1 + \frac{1}{2} \cos(2\omega_0 t) . \end{aligned} \quad (4.11)$$

Plugging this into the expression (4.10), we obtain:

$$\begin{aligned} \mathcal{E}_R &= \int_0^T \left( 1 + \frac{1}{2} \cos(2\omega_0 t) \right) dt \\ &= T + \frac{\sin(2\omega_0 T)}{4\omega_0} \\ &\stackrel{(1)}{\approx} T , \end{aligned} \quad (4.12)$$

where (1) is valid for high frequencies  $f_0$ , since  $-1 \leq \sin(\cdot) \leq 1$  implies  $\sin(\cdot)/(4\omega_0) = \sin(\cdot)/(8\pi f_0) \rightarrow 0$ .

Therefore, on average, the adversary only increases the energy of the resulting signal  $r(t)$ ; the energy content of  $r(t)$  without the adversary is  $T/2$  (Figure 4.10)!

### Gaussian Distribution of $\Theta$

It is reasonable to assume that the adversary cannot perfectly estimate the distances between himself and both the sender and the receiver. This imperfection can be captured by considering the distance shift  $\Delta d$  to be a random variable, i.e., we can assume  $\Delta d$  to be a Gaussian random variable with zero mean and variance  $\sigma_d^2$ . From the expression (4.5),  $\Theta$  is also a Gaussian random variable with zero mean and variance  $\sigma_{\theta}^2 = (2\pi f_0/c)^2 \sigma_d^2$ . To calculate the energy content of  $R(t)$ , we proceed as in the case of the uniform distribution.

$$\begin{aligned} E [R^2(t)] &= \int_{-\infty}^{\infty} r^2(t) f_{\Theta}(\theta) d\theta \\ &= \int_{-\infty}^{\infty} (\cos(\omega_0 t) - \cos(\omega_0 t - \theta))^2 \\ &\quad \times \frac{1}{\sqrt{2\pi}\sigma_{\theta}} e^{-\theta^2/(2\sigma_{\theta}^2)} d\theta \end{aligned} \quad (4.13)$$

By plugging  $E [R^2(t)]$  in the expression (4.10), we obtain the expression for the energy content of the random process  $R(t)$ , with  $\Theta$  being the Gaussian variable. On Figure 4.13 we plot the resulting

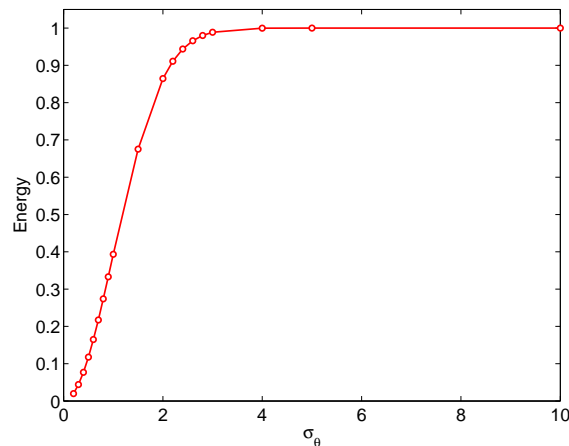


Figure 4.13: The energy of  $R(t)$  (normalized to  $T$ ) for  $\Theta$  Gaussian variable with variance  $\sigma_\theta^2$ .

values of the energy as a function of  $\sigma_\theta$ , for  $f_0 = 5$  GHz. As before, on average, the adversary increases the energy of the resulting signal, except for the low standard deviation  $\sigma_\theta = 1.189$  rad; note that this corresponds to  $\sigma_d = \sigma_\theta / (2\pi f_0) = 1.14$  cm. In addition, the adversary “only” halves the energy of the original signal  $s(t)$  for  $\sigma_\theta = 0.7578$  rad; this value corresponds to  $\sigma_d = 7.236$  mm.

From the analysis in this section, we conclude that we can easily ensure that the adversary cannot block the symbol “1” emitted over a radio channel, even under very advantageous assumptions for him (i.e., no multipath fading effects, perfect estimate of signal amplitudes, etc.).

## 4.7 Related Work

Providing integrity and authentication over insecure (radio) channels is a very active area of research. This provision has mainly focused on the key establishment after which the integrity and the authenticity of the messages is ensured by the use of known cryptographic techniques.

We have already referred to a number of related studies in Section 3.7 (Chapter 3).

## 4.8 Summary

In this chapter, we have introduced *integrity (I) codes*, a novel coding scheme that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (i.e. public keys or shared secret keys). We have analyzed *I*-codes in detail and we have shown that they are secure in a realistic attacker model.

We have further introduced a novel mechanism, called *authentication through presence* based on *I*-codes. We demonstrated the use of this mechanism in two application scenarios: broadcast authentication and key establishment.

We implemented *I*-codes on the Mica2 wireless sensor platform. We demonstrated that *I*-codes can be implemented efficiently and without the use of any specialized hardware.





# Conclusion

In this thesis, we made several original contributions. We began by addressing the problem of cheating in single collision domain CSMA/CA networks. In this context, we provided several contributions. First, we proposed a formalism for the systematic study of rational cheating in CSMA/CA networks, which is based on game theory. Second, we studied the simple cases (i) of a single cheater and (ii) of several cheaters acting without restraint. Third, we showed that the Nash Bargaining Framework (and the Nash Bargaining Solution) is applicable and a useful tool to address resource allocation problems on the MAC layer of wireless networks, even in the face of non-convexity and non-compactness of feasible payoff sets. Using the Nash bargaining framework, we identified the Pareto optimal point of operation of a network with multiple cheaters. Fourth, using the theory of repeated (multistage) games, we showed how it is possible to transform the Pareto optimal point into a Subgame Perfect Nash Equilibrium. Fifth, we showed that smart cheaters can collectively find this point. We believe these contributions to be very relevant in self-organized settings with selfish users.

In the context of radio jamming against wireless networks, we introduced the notion of *coverage paradox* – describing that in spite of the fact that an event is sensed by one or several nodes (and the sensor network is fully connected), the network operator cannot be informed on time. We then showed how these attacks can be thwarted by means of *probabilistic wormholes* – the on-demand (reactive) mechanism ensuring timely delivery of important information. In this thesis, we proposed three realistic wormhole defense mechanisms based on (i) *wired pairs of sensor nodes*, (ii) *coordinated frequency-hopping pairs*, and (iii) *uncoordinated channel-hopping*.

We developed appropriate mathematical models for two solutions, namely, wired and frequency-hopping pairs of sensor nodes. Furthermore, we quantified the probability of success in all the three “probabilistic wormholes”-based approaches. We showed that the approach based on uncoordinated channel-hopping is a particularly well suited defense mechanism for wireless sensor networks.

Concerning the fundamental problem of key agreement over an insecure (radio) link, we made several contributions. First, we proposed a novel and re-usable *MT-authenticator* (MT-SC) based on string comparison, by which users can optimally trade-off the desired security with their involvement in the protocol execution. We showed how the MT-SC authenticator can be used in a modular way to build secure key agreement protocols in the setting where users share no prior secret or certified information. All that users have to do is to compare a short authentication string. In this context, we proposed a novel Diffie-Hellman based key agreement protocol (DH-SC). We proved its security using MT-SC authenticator as a basic building block. Second, we introduced a novel security property called the *integrity region* in the context of the distance bounding based Diffie-Hellman key agreement protocol. Third, we introduced *integrity (I) codes*, a novel coding scheme that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (i.e. public keys or shared secret keys). We analyzed *I*-codes in detail and we showed that they are secure in a realistic attacker model. Fourth, we introduced a novel mechanism,

called *authentication through presence*, which is based on *I*-codes. We demonstrated the use of this mechanism in two application scenarios: broadcast authentication and key establishment. Fifth, we implemented *I*-codes on the Mica2 wireless sensor platform. We demonstrated that *I*-codes can be implemented efficiently and without the use of any specialized hardware.

Last, but not least, in Appendix A, we present some relevant results achieved in the context of the problem of finding power-efficient broadcast trees in wireless networks. We provided novel contributions on several relevant aspects of power-efficient broadcasting in all-wireless networks. First, we studied the complexity of the problem: we discussed two configurations, represented each by a specific graph - a general graph and a graph in Euclidean space (geometric case). For both, we showed that the problem is NP-complete. Furthermore, we showed that the general version cannot be approximated better than  $O(\log N)$ . Second, we elaborated an approximation algorithm for the general version that achieves an approximation ratio of  $18 \log N$ . Also, we elaborated a new algorithm called Embedded Wireless Multicast Advantage (EWMA) that compares well with the existing proposals.

We believe these contributions to be relevant in emerging self-organized wireless networks.

## Directions for Future Work

Concerning selfish behavior on the MAC layer, we envision extending the game theoretic model used for a single collision domain to wireless networks of general topology. We believe this to be a right approach to defining appropriate fairness metrics for self-organized wireless networks.

Related to anti-jamming techniques in the context of wireless sensor networks, we would like to develop an appropriate mathematical model for the approach based on uncoordinated channel hopping. It would be also interesting to evaluate the performances of hybrid solutions, obtained by combining the three approaches proposed in Chapter 2. Finally, it would be interesting to implement the presented schemes.

In the context of secure key agreement mechanisms, we would like to extend the proposed protocols to multiparty settings, that is, to a group key agreement. Also, it would be interesting to implement the distance bounding based key agreement protocol.

## Appendix A

# Minimum-Energy Broadcasting in All-Wireless Networks

During my PhD work, I also studied the problem of constructing *minimum-energy broadcasting trees* in static wireless networks (e.g., sensor networks). In this appendix, I describe relevant results I achieved in this context.

### A.1 Introduction

An all-wireless network consists of numerous devices (nodes) that are equipped with processing, memory and wireless communication capabilities, and are linked via short-range ad hoc radio connections. This kind of network has no pre-installed infrastructure, but all communication is supported by multi-hop transmissions, where intermediate nodes relay packets between communicating parties. Each node in such a network has a limited energy resource (battery) and operates unattended. Consequently, energy efficiency is an important design consideration for these networks [97, 109].

In this chapter, we focus on the source-initiated broadcasting of data in static all-wireless networks. Data are distributed from a source node to each node in a network. Our main objective is to construct a *minimum-energy broadcast tree* rooted at the source node. Nodes belonging to a broadcast tree can be divided into two categories: *relay* nodes and *leaf* nodes. The relay nodes are those that relay data by forwarding it to other nodes (relaying or leaf), and leaf nodes only receive data. Each node can transmit at different power levels and thus reach a different number of neighboring nodes. Given the source node  $r$ , we want to find a set of relaying nodes and their respective transmission levels so that all nodes in the network receive a message sent by  $r$ , whereby the total energy expenditure for this task is minimized. We call this broadcasting problem the *minimum-energy broadcast* problem.

We base our work on the so called *node-based* multicast model [108]. In this model there is a trade-off between reaching more nodes in a single hop thus using more energy and reaching fewer nodes using less energy. This trade-off is made possible by the *broadcast nature* of the wireless channel.

The rest of the chapter is organized as follows. In Section A.2, we discuss the system model used. In Section A.3, we prove that the minimum-energy broadcast problem is NP-complete and show that it cannot be approximated better than  $O(\log N)$  for a general graph, where  $N$  is the number of nodes in a network; we also give the NP-completeness result for the geometric version of the minimum-

energy broadcast problem. Then, in Section A.4, we present  $O(\log N)$ -approximation algorithms for the general graph version and a heuristic algorithm that is easy to distribute. Performance evaluation results are presented in Section A.5. In Section A.6, we overview related work concerning the minimum-energy broadcast problem. Finally, we summarize the results in Section A.7.

## A.2 System Model

We first provide a model of wireless communications. Then using it as a basis, we develop a graph model, which will be used to assess the complexity of the minimum-energy broadcast problem and to develop an approximation algorithm.

In our model of a wireless network, nodes are stationary. In this paper, we assume a large availability of bandwidth resources, i.e. communication channels. We do so because we focus only on minimum energy broadcast communication and do not consider issues like contention for the channel, lack of bandwidth resources. We also assume that nodes in a network are equipped with omnidirectional antennas. Thus due to the broadcast nature of wireless channels, all nodes that fall in the transmission range of a transmitting node can receive its transmission. This property of wireless media is called *Wireless Multicast Advantage*, which we refer to as WMA [108].

In this model, each node can choose to transmit at different power levels that do not exceed some maximum value  $p_{\max}$ . Let  $P$  denote the set of power levels at which a node can transmit. When a node  $i$  transmits at some power level  $p \in P$ , we assign it a weight equal to  $p$ , which we call a *node power*. The connectivity of the network depends on the transmission power. Node  $i$  is said to be *connected* to node  $j$  if node  $j$  falls in the transmission range of node  $i$ . This link is then assigned a *link cost*  $c_{ij}$  that is equal to the minimum power that is necessary to sustain link  $(i, j)$ .

Next we define a graph model for wireless networks, which captures important properties of wireless media (including the wireless multicast advantage). An all-wireless network can be modeled by a directed graph  $G = (V, E)$ , where  $V$  represents the finite set of nodes and  $E$  the set of communication links between the nodes. Each edge (arc)  $(i, j) \in E$  has link cost  $c_{ij} \in \mathbb{R}_+$  assigned to it, and each node  $i \in V$  is assigned a *variable node power*  $p_i^v$ . The variable node power takes a value from the set  $P$  defined above. Initially, the variable node power assigned to a node is equal to zero and is set to value  $p \in P$  if the node transmits at  $p$ . Let  $V_i$  denote the set of *neighbors* of node  $i$ . Node  $j$  is said to be a *neighbor* of node  $i$  if node  $j$  falls in the maximum transmission range of node  $i$ , which is determined by  $p_{\max}$ . All nodes  $j \in V_i$  that satisfy  $c_{ij} \leq p_i^v$  are said to be *covered* by node  $i$ . Thus, if node  $i$  transmits at power  $p_{\max}$ , all the nodes of  $V_i$  will be covered.

Now that we have the model, we study in detail the intrinsic complexity of the minimum-energy broadcast problem in the following section.

## A.3 Complexity Issues

In this section, we give an in-depth analysis of the complexity of the minimum-energy broadcast problem. Let us first briefly recall a few concepts from complexity theory [42]. The problems polynomially solvable by *deterministic* algorithms belong to the P class. Whereas, all the problems solvable by *nondeterministic* algorithms belong to the NP class. It can easily be shown that  $P \subseteq NP$ . Also, there is widespread belief that  $P \neq NP$ . The theory of complexity is focused on *decision problems*, i.e., problems that have either *yes* or *no* as an answer. Notice that each optimization problem can be easily stated as the corresponding decision problem. Informally, a decision problem  $\Pi$  is said to be NP-complete if  $\Pi \in NP$  and for all other problems  $\Pi' \in NP$ , there exists a polynomial

transformation from  $\Pi'$  to  $\Pi$  (we write  $\Pi' \propto \Pi$ ) [42]. There are two important properties of the NP-complete class. If any NP-complete problem could be solved in polynomial time, then all problems in NP could also be solved in polynomial time. If any problem in NP is intractable<sup>1</sup>, then so are all NP-complete problems. Presently, there is a large collection of problems considered to be intractable.

In this section, we consider the problem of minimum-energy broadcast in two different graph models, specifically a general graph and a graph in Euclidean metric space. In general graphs, links are arbitrarily distributed and have weights arbitrarily chosen from the set  $P$ . This graph model is well suited for modeling wireless networks in indoor environments. Whereas, for graphs in Euclidean metric space, the existence and the weight of the link between two nodes depends exclusively on the distance between the nodes and their transmission levels. This graph model fits well for outdoor scenarios.

### A.3.1 General Graph Version

In the following, we show that a general graph version of the minimum-energy broadcast problem is intractable, that is, it belongs to the NP-complete class. Because of its similarity to the well known *Set Cover* problem [48] that aims at finding the minimum cost cover for a given set of nodes, we call it the *Minimum Broadcast Cover* and refer to it as MBC. We convert MBC into a decision problem in the following way:

MINIMUM BROADCAST COVER (MBC)

INSTANCE: A directed graph  $G = (V, E)$ , a set  $P$  consisting of all power levels at which a node can transmit, edge costs  $c_{ij} : E(G) \rightarrow \mathbb{R}_+$ , a source node  $r \in V$ , an assignment operation  $p_i^v : V(G) \rightarrow P$  and some constant  $B \in \mathbb{R}_+$ .

QUESTION: Is there a node power assignment vector  $A = [p_1^v \ p_2^v \ \dots \ p_{|V|}^v]$  such that it induces the directed graph  $G' = (V, E')$ , where  $E' = \{(i, j) \in E : c_{ij} \leq p_i^v\}$ , in which there is a path from  $r$  to any node of  $V$  (all nodes are covered), and such that  $\sum_{i \in V} p_i^v \leq B$ ?

Notice that the above question is the equivalent of asking if there is a broadcast tree rooted at  $r$  with total cost  $B$  or less, and such that all nodes in  $V$  are included in the tree (covered).

We prove NP-completeness of MBC for a general graph by showing that a special case of MBC is NP-complete. In order to obtain this special case of MBC, we define the following restriction to be placed on the instances of MBC: All the links between any node  $i$  and its neighbors  $j \in V_i$  have the same cost  $c$ . Consequently, the node  $i$  either does not transmit or it transmits with  $p_i^v = c$ . We call this special case SINGLE POWER MBC. We prove NP-completeness of the SINGLE POWER MBC problem by reduction from the SET COVER (SC) problem, which is well known to be NP-complete [42].

SET COVER (SC)

INSTANCE: A set  $I$  of  $m$  elements to be covered and a collection of sets  $S_j \in I$ ,  $j \in J = \{1, \dots, n\}$ . Weights  $w_j$  for each  $j \in J$ , and a constant  $B \in \mathbb{R}^+$ .

QUESTION: Is there a subcollection of sets  $C$  that form a cover, i.e.,  $\cup_{j \in C} S_j = I$  and such that  $\sum_{j \in C} w_j \leq B$ ?

First we describe the construction of a graph  $G$  that represents any instance of the set cover problem. The graph  $G$  has a vertex set  $I \cup \{v_1, v_2, \dots, v_n\}$ , that is,  $G$  consists of elements of  $I$  and set vertices  $v_j$  representing sets  $S_j \in I$ ,  $j \in J = \{1, \dots, n\}$ . There is an edge between an element  $e \in I$  and a set node  $v_i$  if the set  $S_i$  contains the element. Each set node  $v_i$  is assigned the weight

<sup>1</sup>We refer to a problem as intractable if no polynomial time algorithm can possibly solve it.

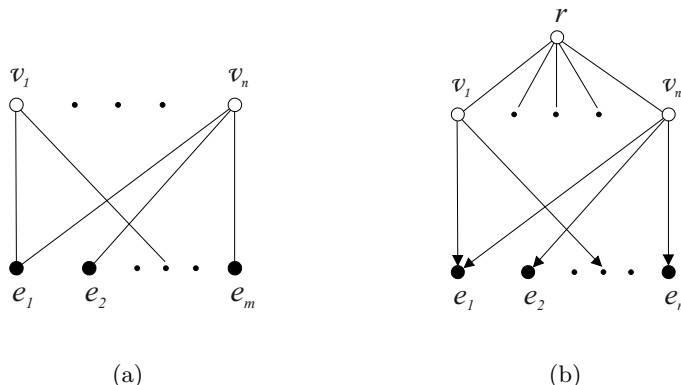


Figure A.1: The reduction of (a) SET COVER to (b) SINGLE POWER MINIMUM BROADCAST COVER

$w_i$  of the set  $S_i$  the node represents. All other nodes and all edges are not weighted, that is, they have weight of zero. Thus,  $G = (V, E)$  is a bipartite graph, as is illustrated in Figure A.1(a).

The transformation from SC to SINGLE POWER MBC consists first in adding a source (root) node  $r$  to  $G$  and making it adjacent to all the set nodes  $v_j$ . Note that we use undirected edges here to emphasize that the links between the source  $r$  and nodes  $v_j$  are bidirectional. We proceed by assigning a zero weight to every edge the root node  $r$  shares with the set nodes  $v_j$ . Then, the edges between  $v_j$  and elements  $e \in I$  are made directed in order to capture the fact that no element  $e \in I$  is ever selected into the cover set  $C$ . Finally, the directed edges the node  $v_j$  shares with elements  $e \in I$  are assigned the weight  $w_j$ . The resulting graph, which we denote with  $G_b = (V_b, E_b)$ , is illustrated in Figure A.1(b). It is easy to see that the transformation can be done in polynomial time.

Next we prove the following theorem.

**Theorem 15** *SINGLE POWER MBC is NP-complete.*

*Proof:* The proof consists first in showing that SINGLE POWER MBC belongs to the NP class and then in showing that the above polynomial transformation (Figure A.1) reduces SC to SINGLE POWER MBC.

It is easy to see that SINGLE POWER MBC belongs to the NP class since a nondeterministic algorithm needs only to guess a set of transmitting nodes ( $p_i^v > 0$ ) and to check in polynomial time whether there is a path from the source node  $r$  to any node in a final solution and whether the cost of the final solution is  $\leq B$ .

We continue the proof by showing that given the minimum broadcast cover  $C_b$  of  $G_b$  with cost  $cost(C_b)$ , the set  $C_b - \{r\}$  always corresponds to the minimum set cover  $C$  of  $G$  of the same cost ( $cost(C) = cost(C_b)$ ), and vice versa. Let  $C$  denote the minimum set cover of  $G$ . Let  $cost(C) = \sum_{j \in C} w_j$  denote the cost of this cover. It is easy to see that all nodes of  $G_b$  can also be covered with total cost  $cost(C)$ . This can be achieved by having the source node  $r$  cover all the set nodes  $v_j$ ,  $j \in J = \{1, \dots, n\}$  at zero cost and then by selecting among the covered nodes those corresponding to the nodes of  $G$  that satisfy  $v_j \in C$  as new transmitting nodes, which we refer to as  $C_b - \{r\}$ . Hence the minimum broadcast cover of  $G_b$  is  $C_b$  with total cost  $cost(C_b) = cost(C)$ .

Conversely, suppose that we have the minimum broadcast cover  $C_b$  of  $G_b$  with total cost  $cost(C_b)$ . Thus the minimum set cover  $C$  of  $G$  must be  $C = C_b - \{r\}$ , i.e.,  $cost(C) = cost(C_b)$ . We prove this by

contradiction. Let us first assume that  $\text{cost}(C) < \text{cost}(C_b)$  (hence  $C \neq C_b - \{r\}$ ). In this case, with the same reasoning as before,  $G_b$  can be covered by  $C'_b = C + \{r\}$  that satisfies  $\text{cost}(C'_b) < \text{cost}(C_b)$ . This, however, contradicts the preceding assumption that  $C_b$  is the minimum broadcast cover of  $G_b$ . On the other hand, let us assume that  $\text{cost}(C_b) < \text{cost}(C)$  (hence  $C \neq C_b - \{r\}$ ). Since  $C_b$  covers all the elements  $e \in I$ , we can obtain a set cover  $C'$  for this instance as follows:  $C' = C_b - \{r\}$ . Now we have  $\text{cost}(C') = \text{cost}(C_b) < \text{cost}(C)$ , which contradicts the optimality of  $C$  and concludes the proof.  $\square$

Since the SINGLE POWER MBC problem is a special case of the MBC problem, and MBC belongs to the NP class, which can be shown along the similar lines as for THE SINGLE POWER MBC problem, we have the following corollary:

**Corollary 2** *MINIMUM BROADCAST COVER (MBC) is NP-complete.*

Another important implication of Theorem 15 is the following theorem. Let  $N$  denote the total number of nodes in an instance of MBC.

**Theorem 16** *There exists a constant  $c > 0$  such that MINIMUM BROADCAST COVER (MBC) cannot be approximate better than  $c \log N$ , if  $P \neq NP$ .*

*Proof:* To prove this we recall that there exists a constant  $c' > 0$  such that no polynomial-time approximation algorithm for SC achieves an approximation ratio smaller than  $c' \log n$  if  $P \neq NP$ , where  $n$  is the total number of elements in an instance of SC [60]. We showed above how any instance of the SC problem can be transformed to the corresponding instance of MBC. Now, assume that we have an approximation algorithm for MBC with the performance guarantee better than  $c' \log(N - 1)$ . By applying this algorithm to the instance of MBC obtained from the SC instance, we would get a solution with a cost lower than  $c' \log(n + 1 - 1) \cdot OPT = c' \log n \cdot OPT$ . Since this solution is also feasible to the instance of SC, this would mean that we can approximate SC better than  $c' \log n$ , which contradicts the fact that SC is hard to approximate better than  $c' \log n$ . We obtain the theorem by noting that  $c' \log(N - 1) = c \log N$ , where  $0 < c \leq c'$  for  $N > 2$ .  $\square$

Fortunately, Theorem 16 does not hold for all instances of the minimum-energy broadcast problem. By exploring the geometric structure of the minimum-energy broadcast problem, Wan et al. were able to show that the Euclidean minimum spanning tree approximates the minimum-energy broadcast problem within a factor of 12 [105]. But, whether the geometric instances of the minimum-energy broadcast problem can be solved in polynomial time was left as an open question. We provide an answer in the next subsection.

### A.3.2 Geometric Version

In this section, we show that the minimum-energy broadcast problem in two-dimensional Euclidean metric space is intractable. In metric space, the distance between points (nodes) obeys triangle inequality, that is,  $d_{ij} \leq d_{ik} + d_{kj}$ , where  $d_{xy}$  is the Euclidean distance between nodes  $x$  and  $y$ . We have seen that given the graph version of the minimum-energy broadcast problem we can have arbitrary costs of links between nodes. This is because we did not have to worry about the distances between nodes and all links have been imposed by a given graph. On the contrary, in metric space, links and their respective costs are dictated by the distances between nodes and their transmission energies. The cost  $c_{ij}$  between two nodes  $i$  and  $j$  is given as

$$c_{ij} = kd_{ij}^\alpha \tag{A.1}$$

where  $k \in \mathbb{R}^+$  is constant depending on the environment and  $\alpha$  is a propagation loss exponent that takes values between 2 and 5 [89].

We refer to this instance of the minimum-energy broadcast problem as the *Geometric Minimum Broadcast Cover* (GMBC) problem. The decision problem related to GMBC can be formulated as follows:

GEOMETRIC MINIMUM BROADCAST COVER (GMBC)

INSTANCE: A set of nodes  $V$  in the plane, a set  $P$  consisting of all power levels at which a node can transmit, a constant  $k \in \mathbb{R}_+$ , costs of edges  $c_{ij} = kd_{ij}^\alpha$  where  $d_{ij}$  is the Euclidean distance between  $i$  and  $j$ , a real constant  $\alpha \in [2..4]$ , a source node  $r \in V$ , an assignment operation  $p_i^v : V(G) \rightarrow P$  and some constant  $B \in \mathbb{R}_+$ .

QUESTION: Is there a node power assignment vector  $A = [p_1^v \ p_2^v \ \dots \ p_{|V|}^v]$  such that it induces the directed graph  $G = (V, E)$ , with an edge (arc) directed from node  $i$  to node  $j$  if and only if  $c_{ij} \leq p_i^v$ , in which there is a path from  $r$  to any node of  $V$  (all nodes are covered), and such that  $\sum_{i \in V} p_i^v \leq B$ ?

Given the above formal definition of the geometric version of the minimum-energy broadcast problem, we have the following theorem:

**Theorem 17** *GEOMETRIC MINIMUM BROADCAST COVER (GMBC) is NP-complete.*

The proof of the theorem can be found in [25]. We proved NP-completeness of GMBC by reduction from the PLANAR 3-SAT problem, which is known to be NP-complete [75].

In the following section, we devise approximation algorithms that enable us to find good solutions to the minimum-energy broadcast problem.

## A.4 Proposed Algorithms

In this section, we first present an approximation algorithm that achieves  $O(\log N)$  approximation ratio for any instance of MBC. Then, we elaborate on the algorithm EWMA, designed deliberately for the geometric version of the minimum energy broadcast problem, and we explain how to convert it to a distributed algorithm.

### A.4.1 $O(\log N)$ -approximation Algorithm

The MBC problem can be seen as a special case of the *hitting set problem*. The hitting set problem is defined as follows [48]: Given subsets  $S_1, \dots, S_p$  of a ground set  $E$  and given a nonnegative cost  $c_e$  for every element  $e \in E$ , find a minimum-cost subset  $A \subseteq E$  such that  $A \cap S_i \neq \emptyset$  for every  $i = 1, \dots, p$  (i.e.  $A$  hits every  $S_i$ ).

In our case, we are given a connected graph  $G = (V, E)$  with positive edge costs and a special root node  $r$ . The sets to hit are all  $r$  directed cuts, i.e. the sets of edges of the form  $\delta^-(S) = \{(i, j) \in E : i \notin S, j \in S\}$  where  $S \subseteq V - \{r\}$ . Informally, for any subset of nodes  $S \subseteq V - \{r\}$ , we should have at least one transmitter  $i \notin S$  that covers at least one node  $j \in S$ . It is easy to see that, if this is fulfilled for all  $S \subseteq V - \{r\}$ , we obtain a feasible solution for MBC. Consequently, any set  $S \subseteq V - \{r\}$  that has no edge incoming to it is said to be violated. For simplicity, we will say that  $S$  is not hit while meaning that  $\delta^-(S)$  is not hit. The number of violated sets can be in theory as large as  $2^{|V|-1}$ , i.e. exponential in the total number of nodes. In order to drastically reduce this number, we apply the technique described in [48], where, instead of considering all possible violated sets, we take into account only *minimal violated sets*. Any violated



$S$  is said to be a minimal violated set if there exists no violated  $S'$  with  $S' \subset S$ . The rule we use to calculate minimal violated sets is defined by the following:

**Definition 11** *The minimal violated set is a strongly connected component (i.e. collection of nodes)  $S \subseteq V - \{r\}$  that contains no directed edge incoming to it.*

We next describe an approximation algorithm (Algorithm 1), which achieves an  $O(\log N)$  approximation ratio for MBC. Let  $C$  denote the set comprising pairs  $(i, k)$  where  $i \in V$  is *transmitter* and where  $k \in P$  its respective *transmission power level*. The algorithm iteratively selects the most cost-effective pair  $(i, k)$  and puts it into the set  $C$  and updates correspondingly the collection of minimal violated sets  $\mathcal{V}$ , until  $\mathcal{V}$  is empty (i.e.  $C$  is a feasible solution).

---

**Algorithm 1**  $O(\log N)$ -approximation algorithm

---

```

1   $C \leftarrow \emptyset; t = 0$ 
2  While  $C$  is not feasible
3     $t \leftarrow t + 1; \mathcal{V}(t) \leftarrow Violation(C)$ 
4     $(i, k)_t = \arg \min_{(i, k)} \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|}$ 
5     $C \leftarrow (i, k)_t; \underline{c}_i(t) = c_i^k(t)$ 
6    For all  $S \in S(t)$ 
7       $price(S) = \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|}$ 
8  For  $j \leftarrow t$  downto 1
9    if  $C - \{(i, k)_j\}$  is feasible  $C \leftarrow C - \{(i, k)_j\}$ 

```

---

At the beginning of the algorithm, the set  $C$  is empty (and thus not feasible). Let  $Violation(C)$  be an oracle that calculates the minimal violated sets of the graph  $G$  for a given  $C$ ; the oracle does this by following Definition 11. The set  $\mathcal{V}(t)$  holds all the minimal violated sets returned by the oracle at the beginning of each iteration  $t$  (line 3). Note that at the very beginning, the number of minimal violated sets is  $N - 1$  (i.e. all  $i \in V - \{r\}$ ). The algorithm selects the most cost-effective pair  $(i, k)$  (line 4); here,  $c_i^k$  denotes the cost assigned to the node  $i$  that transmits at the power level  $k$  and  $\underline{c}_i$  represents the cost induced by any previous selection of the node  $i$  into the set  $C$ . Thus, we allow a node to be selected more than one time in the final solution, which does not mean that the node actually transmits two times or more. This uniqueness is ensured by the delete step (lines 8 and 9). The set  $S(t) \subseteq \mathcal{V}(t)$  is defined as follows  $S(t) = \{S \in \mathcal{V}(t) : \delta^-(S) \text{ is hit by } (i, k)_t\}$ . Informally,  $S(t)$  comprises the minimal violated sets that are newly hit at the iteration  $t$ . The sets  $S \in S(t)$  are then assigned the price (line 7), which will be used in the proof of Theorem 18.

In the rest of this section we evaluate the performance guarantee of the algorithm. Let  $A_t$  denote the event that a new minimal violated set is induced in iteration  $t$ . We first prove the following lemma:

**Lemma 4**

$$|\mathcal{V}(t+1)| = \begin{cases} |\mathcal{V}(t)| - |S(t)| + 1, & \text{if } A_t \\ |\mathcal{V}(t)| - |S(t)|, & \text{otherwise.} \end{cases}$$

*Proof:* Let us consider the iteration  $t$  with  $|\mathcal{V}(t)|$  minimal violated sets. Let  $(i, k)_t$  be selected into the set  $C$  at this iteration, that is,  $(i, k)_t$  hits at least one minimal violated set  $S \in \mathcal{V}(t)$ . Then we have the following two possibilities: either by this transmission  $i$  produces no new minimal violated set (strongly connected component) or it produces one or more. Clearly, in the first case we have  $|\mathcal{V}(t+1)| = |\mathcal{V}(t)| - |S(t)|$ .

In the second case, the node  $i$  is included in any strongly connected component (minimal violated set) that it has newly produced. Consequently, there must exist a directed path from every node of such components to the node  $i$ , and vice versa, from the node  $i$  to any node of these components. This in turn means that these components belong to the same strongly connected component (i.e. the same minimal violated set). Therefore,  $i$  induces, at most, one new minimal violated set, in which case  $|\mathcal{V}(t+1)| = |\mathcal{V}(t)| - |S(t)| + 1$ .  $\square$

Let us introduce the following indicator variable:

$$I(t) = \begin{cases} 1, & \text{if in } t \text{ a new violated set is induced} \\ 0, & \text{otherwise.} \end{cases}$$

Let  $m$  denote the total number of iterations of our algorithm, and  $l$  the total number of minimal violated sets during the course of the algorithm. Then by using Lemma 4 and observing that  $|\mathcal{V}(1)| = N - 1$  and  $|\mathcal{V}(m+1)| = 0$ , we obtain:  $l = \sum_{t=1}^m |S(t)| = \sum_{t=1}^m I(t) + N - 1$ .

We next evaluate the bound on the total number of newly generated minimal violated sets.

**Lemma 5**

$$\sum_{t=1}^m I(t) \leq N - 2$$

*Proof:* By Definition 11, every newly created minimal violated set is a strongly connected component. Therefore,  $\sum_{t=1}^m I(t)$  is, at most, the number of newly generated strongly connected components. At the very beginning, the number of eligible nodes (components) for the creation of newly strongly connected components is  $N - 1$ . Since each time a new strongly connected component is created at least two eligible components are merged, the number of eligible components is decreased by at least 1. Therefore, the total number of newly created strongly connected components is at most  $N - 2$ , which concludes the proof.  $\square$

By applying Lemma 5 to the expression for the total number of minimal violated sets  $l$ , we obtain:  $l = \sum_{t=1}^m |S(t)| \leq 2N - 3$ . We can use this inequality to obtain the upper bound on the total number of iterations  $m$ . Having  $|S(t)| = 1$  in every iteration  $t$ , we obtain:  $m \leq 2N - 3$ . Since the violation oracle can be implemented to run in polynomial time<sup>2</sup>, our algorithm is polynomial in the total number of nodes  $N$ .

Let  $OPT$  denote the total cost of the optimal solution. We next prove the following lemma, which is similar to Lemma 2.3 in [103]:

**Lemma 6** For each  $(i, k)_t$  selected into  $C$ ,  $\frac{c_i^k(t) - c_i(t)}{|S(t)|} \leq \frac{OPT}{|\mathcal{V}(t)|}$ .

*Proof:* In any iteration  $t$ , transmitters from the optimal solution can cover the sets from  $\mathcal{V}(t)$  at a cost of at most  $OPT$ . Consequently, the cost-effectiveness of any of these transmitters is at

---

<sup>2</sup>For example STRONGLY CONNECTED COMPONENT ALGORITHM given in [60] runs in  $O(N + |E|)$ , where  $|E| < N^2$  (Theorem 2.19).

most  $\frac{OPT}{|\mathcal{V}(t)|}$ . Therefore, by selecting the most cost-effective  $(i, k)_t$  at the iteration  $t$  (i.e.  $(i, k)_t = \arg \min_{(i, k)} \frac{c_i^k(t) - c_i(t)}{|S(t)|}$ ), we must have  $\frac{c_i^k(t) - c_i(t)}{|S(t)|} \leq \frac{OPT}{|\mathcal{V}(t)|}$ .  $\square$

Finally, we prove the following theorem on the performance guarantee of our approximation algorithm.

**Theorem 18** *Algorithm 1 delivers a feasible solution of cost not larger than  $c \log N \cdot OPT$ , where  $c = 2 \cdot \left( \frac{1}{\log e} + \frac{2}{\log N} \right) < 18$  for any  $N \geq 2$ . That is, Algorithm 1 is an  $O(\log N)$ -approximation algorithm<sup>3</sup>.*

*Proof:* Since the cost of each pair  $(i, k)_t$  of the output  $C$  is evenly distributed among the newly hit minimal violated sets  $S(t)$ ,  $cost(C) = \sum_{j=1}^l price(S_j) = \sum_{t=1}^m \sum_{j=1}^{|S(t)|} price(S_j)$ . Now we have:

$$cost(C) = \sum_{t=1}^m \sum_{j=1}^{|S(t)|} \frac{c_i^k(t) - c_i(t)}{|S(t)|} \quad (\text{A.2})$$

$$\leq \sum_{t=1}^m \sum_{j=1}^{|S(t)|} \frac{OPT}{|\mathcal{V}(t)|} \quad (\text{A.3})$$

$$= 2 \cdot OPT \sum_{t=1}^m \sum_{j=1}^{|S(t)|} \frac{1}{2|\mathcal{V}(t)|} \quad (\text{A.4})$$

$$\leq 2 \cdot OPT \sum_{t=1}^m \sum_{j=1}^{|S(t)|} \frac{1}{2|\mathcal{V}(t)| - j + 1} \quad (\text{A.5})$$

$$< 2 \cdot OPT \sum_{i=1}^{2(N-1)} \frac{1}{i} \quad (\text{A.6})$$

$$\leq 2 \cdot [\ln(N-1) + \ln 2 + 1] \cdot OPT \quad (\text{A.7})$$

$$< 2 \cdot \left( \frac{1}{\log e} + \frac{2}{\log N} \right) \cdot \log N \cdot OPT \quad (\text{A.8})$$

where (A.3) follows from Lemma 6; (A.6) follows from Lemma 4,  $\sum_{t=1}^m |S(t)| < 2(N-1)$  and  $|\mathcal{V}(1)| = N-1$ ; and (A.7) follows from the inequality  $\sum_{j=1}^n \frac{1}{j} \leq \ln n + 1$ .  $\square$

In this subsection we developed the approximation algorithm for the general graph version (MBC). In the following two subsections, we first elaborate on a centralized heuristic algorithm deliberately designed for the geometric version (GMBC). Then, we explain how it can be converted to a distributed algorithm.

#### A.4.2 A Heuristic Based Approach

Let us first present an informal description of the heuristic we propose. We first construct a feasible solution (an initial feasible broadcast tree). Then we improve this solution by exchanging some existing branches in the initial tree for new branches so that the total energy necessary to maintain the broadcast tree is reduced. We do it so that the feasibility of the obtained solution remains intact. We call the difference in the total energies of the trees before and after the branch exchange

---

<sup>3</sup>log designates base 10 logarithm.

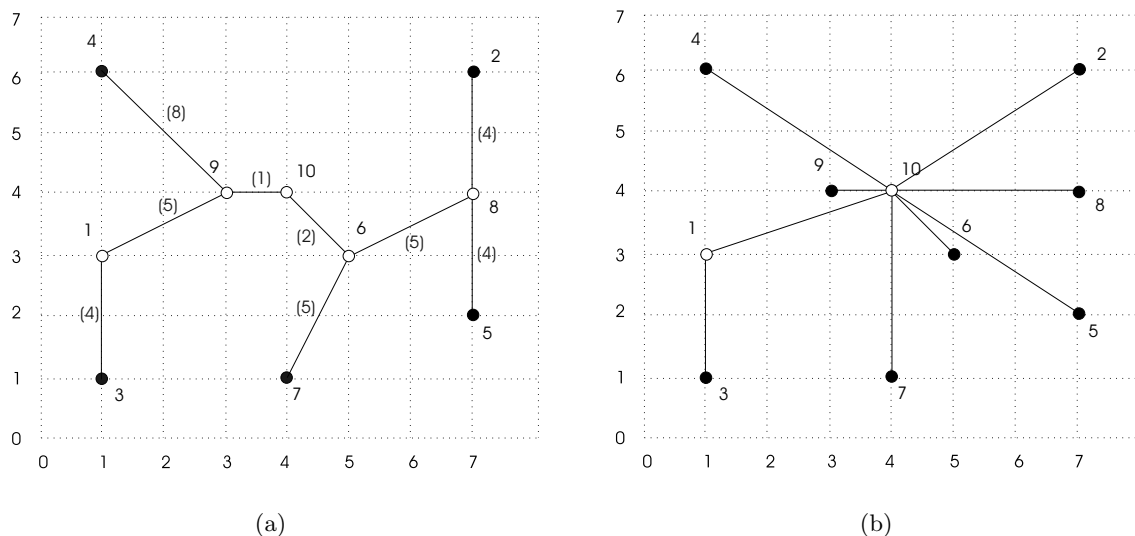


Figure A.2: Example of the EWMA algorithm: (a) the initial MST ( $e_{MST} = 23$ ) and (b) the broadcast tree obtained by EWMA ( $e_{EWMA} = 17$ )

a *gain*. In our heuristic, the notion of gain is used as the criterion for the selection of transmitting nodes in the broadcast tree.

We use the link-based minimum spanning tree (MST) as the initial feasible solution. The main reason we choose MST is that it performs quite well, even as a final solution to our problem (which can be seen from the simulation results in Section A.5).

We will now describe in detail our algorithm, which we call *Embedded Wireless Multicast Advantage* (EWMA). An example is provided in Figure A.2. Let us first introduce some notations. Let  $C$  denote the set of covered nodes,  $F$  the set of transmitting nodes of the final broadcast tree, and  $E$  the set of *excluded* nodes. Node  $i$  is said to be an *excluded node* if is transmitting node in the initial solution but not in the final solution (i.e.  $i \notin F$ ). Notice that the contents of the above sets change throughout the execution of the EWMA and that the sets do not hold any information about the MST. Initially,  $C = \{r\}$ , where  $r$  is the source node (node 10 in our example), and sets  $F$  and  $E$  are empty.

In this example, we assume a propagation loss exponent  $\alpha = 2$ . After the MST has been built in the *initialization* phase, we know which nodes in the MST are transmitting nodes and their respective transmission energies. In our example, the transmitting nodes are 10, 9, 6, 1, 8, and their transmission energies are 2, 8, 5, 4, and 4, respectively. The total energy of the MST is  $e_{MST} = 23$ . Notice here that we take into consideration the wireless multicast advantage in the evaluation of the cost of the MST. Notice also that  $C = \{10\}$ , and  $F = E = \{\emptyset\}$ . In the second phase, EWMA starts to build a broadcast tree from nodes in the set  $(C - F) - E$  by determining their respective gains. The gain of a node  $v$  is defined as the decrease in the total energy of the broadcast tree obtained by excluding some of the nodes from the set of transmitting nodes in MST, in exchange for the increase in node  $v$ 's transmission energy. Notice that this increase of node  $v$ 's transmission energy has to be sufficient for it to reach all the nodes that were previously covered by the nodes that were excluded. Consequently, the feasibility of the solution is preserved. At this stage of the

algorithm, the set  $(C - F) - E$  contains only the source node 10. Thus for example, in order to exclude node 8, the source node 10 has to increase its transmission energy by (see Figure A.2):

$$\Delta e_{10}^8 = \max_{i \in \{2,5\}} \{e_{10,i}\} - e_{10} = 13 - 2 = 11$$

The gain ( $g_{10}^8$ ) obtained in this case is:

$$g_{10}^8 = e_6 + e_8 + e_9 - \Delta e_{10}^8 = 5 + 4 + 8 - 11 = 6$$

where  $e_i, i = \{6, 8, 9\}$ , is the energy at which node  $i$  transmits in MST. Notice that, in addition to node 8, the nodes 6 and 9 can also be excluded.

Likewise,  $g_{10}^1 = 5$ ,  $g_{10}^6 = -2$ , and  $g_{10}^9 = 6$ . Having the gains for all nodes from  $(C - F) - E$ , our algorithm selects the node with the highest positive gain in the set  $F$ . Our algorithm then adds all the nodes that this node excludes to the set  $E$ . Thus the source node 10 is selected in the set  $F$  to transmit with energy that maximizes its gain, that is:

$$e'_{10} = e_{10} + \arg \max_{\Delta e_{10}^i} \{g_{10}^i\}, \quad g_{10}^i \geq 0$$

The source node 10 transmits with energy  $e'_{10} = e_{10} + \Delta e_{10}^8 = 2 + 11 = 13$  at which it can cover nodes 6, 8, 9 and all their *child* nodes in MST. Node  $j$  is said to be a *child node* of node  $i$  if node  $j$  is included in the broadcast tree by node  $i$ . Hence, at this stage we have  $C = \{1, 2, 4, 5, 6, 7, 8, 9, 10\}$ ,  $E = \{6, 8, 9\}$  and  $F = \{10\}$ . If none of the nodes from  $(C - F) - E$  has a positive gain, EWMA selects among them the node that includes its child nodes in the MST at minimum cost (energy).

The above procedure is repeated until all nodes in the network are covered. In our example, there is still one node to be covered, namely node 3. Again, EWMA scans the set  $(C - F) - E = \{1, 2, 4, 5, 7\}$  and at last selects node 1 to be the next forwarding node. When node 1 transmits with energy  $e_1 = 4$ , all nodes are covered ( $C = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ ) and the algorithm terminates. At the final stage we have  $E = \{6, 8, 9\}$  and  $F = \{1, 10\}$ . The resulting tree, shown in Figure A.2(b), has a cost  $e_{EWMA} = 17$ . Notice that our algorithm always results in a broadcast tree with a total energy  $\leq e_{MST}$ , which is, in the case of Euclidean MST, at most  $12e_{opt}$  [105].

In the next subsection we explain how to convert our centralized heuristic algorithm to a distributed algorithm.

### A.4.3 Distributed Implementation of EWMA

One of the major research challenges, with respect to the broadcasting problem, is the development of a distributed algorithm [108, 105]. In the following we describe our solution.

Let us first introduce the notations we will be using. Let node  $i$  transmit at power level  $p \in P$ . We denote the set of nodes that are covered by this transmission with  $V_i^p$ . Let node  $j$  be a neighbor of  $i$ , that is,  $j \in V_i$ . We denote with  $O_{ij}^p$  the set of nodes belonging to  $V_i^p \cap V_j$  and call it the *overlapping* set. We assume that each node knows its two-hop neighborhood. So, once node  $j$  receives a message from node  $i$ , it can learn which of the nodes from its neighbor set  $V_j$  have also received the message by calculating the overlapping set  $O_{ij}^p$ . The neighbors of node  $j$  that have not yet received the message are said to be *uncovered*, and we denote this set with  $U_j$  where  $U_j = V_j - O_{ij}^p$ . If node  $j$  is a forwarding node in the MST, then the set of yet uncovered children nodes of node  $j$  in the MST is denoted with  $U_j^{mst}$  where  $U_j^{mst} = V_j^{mst} - O_{ij}^p$ . Here,  $V_j^{mst}$  is the set comprising all the children nodes of  $j$  in the MST. Finally, we denote with  $e_j^{mst}$  the energy with which node  $j$  transmits in the MST.

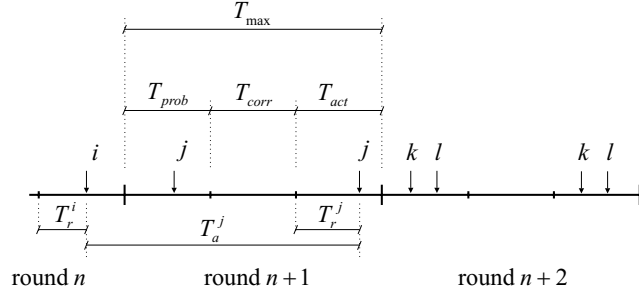


Figure A.3: Synchronization of the second phase of distributed EWMA

Our distributed algorithm is divided into two phases. In the first phase, all nodes run a distributed algorithm proposed by Gallager et al. [41] to construct a *minimum-weight spanning tree*. The total number of messages required for a graph of  $|V|$  nodes and  $|E|$  edges is at most  $5|V| \log_2 |V| + 2|E|$ , and the time until completion is  $O(|V| \log |V|)$  [41]. Notice that Gallager et al. considered the link-based model, whereas we use the node-based multicast model, which captures the wireless multicast advantage property [108]. As a result, the total number of messages required in our model may be considerably lower. We require that at the end of the first phase, each node has information about the cost of its two-hop neighbors related to the MST built.

In the second phase, the final broadcast tree is built up. The main difficulty in this distributed setting is the unavailability of information about which nodes have been covered, up to a certain moment. In order to cope with this problem, we apply two techniques. First, we organize this second phase in rounds. Second, we require that the identities of the nodes on the transmission chain from the source to a given node, along with their respective transmission powers are propagated along that chain to the node in question (source routing technique).

Each round of the second phase is  $T_{\max}$  long. Rounds are additionally divided into three time periods, namely, a *probation* period ( $T_{\text{prob}}$ ), a *correction* period ( $T_{\text{corr}}$ ), and an *active* period ( $T_{\text{act}}$ ), which are all known by network nodes (Figure A.3). Let node  $i$  transmit at  $T_r^i$  time from the beginning of the active period of round  $n$ . Node  $j$  receives this message and begins the following update procedure. It calculates the overlapping set for the sender  $i$  and for other transmitters on this chain of transmitting nodes for which node  $j$  has neighbors in common (recall that this information is propagated along the chain). If node  $j$  is a forwarding node in the MST and it finds that the set of uncovered nodes  $U_j^{\text{mst}}$  is empty for the received message, it will not re-broadcast the message. Otherwise, (i.e. if  $U_j^{\text{mst}}$  is non-empty or  $j$  was a leaf node in the MST), it calculates the gains it can achieve by covering yet uncovered nodes (based on locally available information), and selects the maximum gain  $g_{j \max}$ . In the case  $g_{j \max} > 0$ , node  $j$  can contribute to the decrease of the total cost of the broadcast tree and its transmission energy increases as follows:  $e_j = e_j + \arg \max_{\Delta e_j^l} \{g_j^l\}$ , otherwise ( $g_{j \max} \leq 0$ ) its transmission energy remains unchanged.

At this stage, node  $j$  waits for some time period  $T_a^j$  before possibly re-broadcasting the message. The waiting period is given as follows:

$$T_a^j = T_{\max} + T_r^j - T_r^i$$

where  $T_r^j = \frac{\Delta_1}{g_{j \max}}$  if  $g_{j \max} > 0$ , and  $T_r^j = \Delta_2 \cdot e_j$  if  $g_{j \max} \leq 0$  and  $e_j > 0$ . In the first case the waiting period  $T_a^j$  is reciprocal to the gain, in order to give priority to nodes with higher positive gains over

nodes with lower positive gains. In the second case, the waiting period  $T_a^j$  is proportional to the transmission energy in order to give priority to nodes with lower transmission energies over nodes with higher transmission energies. Additionally, the nodes with positive gains are given priority to the nodes with low transmission energies (i.e.  $\frac{\Delta_1}{g_{j \max}} \ll \Delta_2 \cdot e_j$ ). This property is ensured by setting appropriately the constants  $\Delta_1$  and  $\Delta_2$ .

Since node  $j$  calculates the gains based on only locally available information, in the calculation of the gains, node  $j$  can try to exclude already excluded nodes. In order to prevent this, node  $j$  transmits a probe message during the probation period  $T_{prob}$  of round  $n + 1$ . Note that by knowing  $T_{act}$  and  $T_r^i$  (which  $j$  received from  $i$ ) node  $j$  actually knows when round  $n + 1$  starts. The probe message carries the addresses of all the nodes by exclusion of which node  $j$  attains  $g_{j \max} > 0$ , and it carries the starting time of the correction period. If some of these nodes have already been excluded, they will respond back to node  $j$  during the correction period. Node  $j$  will accordingly update its gain and the waiting period  $T_a^j$  by taking into account the already elapsed time of the waiting period. The duration of the probation and correction periods should be such that any potential forwarding node is given the chance to test its prospect of actually being the forwarding node.

Finally, node  $j$  enters into the active period. Again, based on the knowledge of  $T_{prob}$  and  $T_{corr}$ , node  $j$  knows when the active period of round  $n + 1$  starts. If during that period and before expiration of the waiting period  $T_a^j$  node  $j$  receives a duplicate message, it repeats the update procedure above, otherwise, upon expiration of  $T_a^j$ , it re-broadcasts the message with energy  $e_j$ , stores this value and marks itself as the forwarding node. In our example shown in Figure A.3, node  $j$  decides to be the forwarding node and broadcasts a message at power  $e_j$ . By doing so, it initiates the update procedure at nodes  $k$  and  $l$  that repeat the whole process.

Next we show under which conditions the waiting period  $T_a^j$  expires solely during the active period of round  $n + 1$ . From Figure A.3 we can see that this happens if  $T_a^j$  conforms to the following conditions:

$$\begin{aligned} T_a^j &\geq T_{\max} - T_r^i \\ T_a^j &\leq T_{\max} + T_{act} - T_r^i \end{aligned}$$

From the first inequality and the definition of  $T_a^j$  we obtain that  $T_r^j \geq 0$ , which is always satisfied. Along the same lines, from the second inequality we obtain that  $T_r^j \leq T_{act}$ . Consequently, we define the active period as follows:

$$\begin{aligned} T_{act} &= \max_{j \in F} \{T_r^j\} \\ &= \max_{j \in F} \{\Delta_2 \cdot e_j\} \end{aligned}$$

where the second equality follows from the fact that  $\frac{\Delta_1}{g_{j \max}} \ll \Delta_2 \cdot e_j$ . Now, since we already have decided on  $\Delta_1$  and  $\Delta_2$ , we only have to find the cost of the most expensive edge in the MST. Note that this information can be obtained from the first phase of the algorithm. This, in addition to the appropriate selection of the periods  $T_{prob}$  and  $T_{corr}$ , ensures a synchronous execution of the second phase of the distributed algorithm.

The duration of the second phase is bounded by  $|F| \cdot T_{\max}$ , where  $F$  is the set of the forwarding nodes at the end of the second phase. Thus, at the end of the second phase, the broadcast tree is built (i.e. we have a set of forwarding nodes  $F$  and their respective transmission energies for a given source node). Any subsequent broadcast message originating from the source node can be

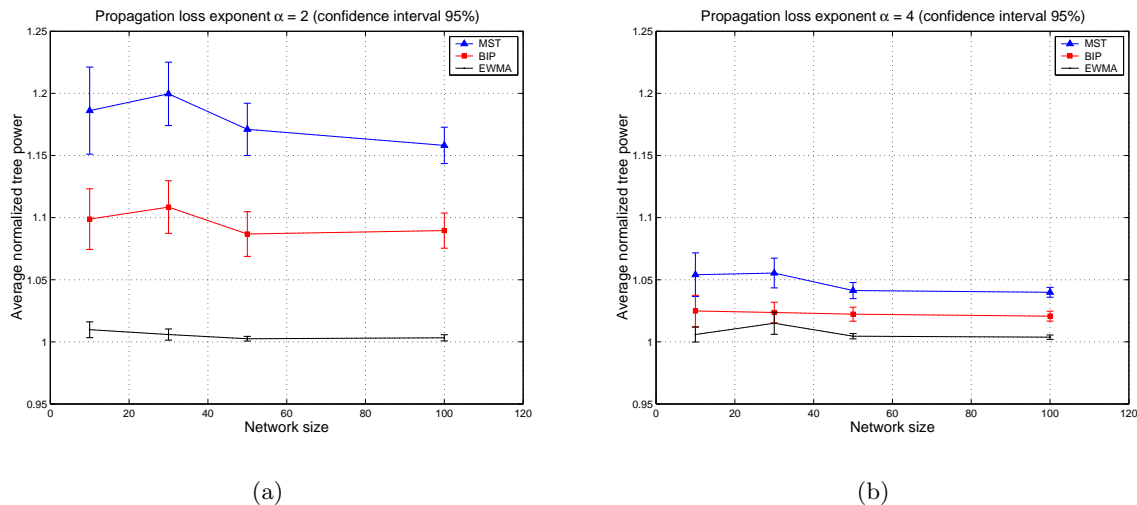


Figure A.4: Normalized tree power comparison: (a)  $\alpha = 2$  and (b)  $\alpha = 4$

disseminated along the tree in an asynchronous way (i.e. forwarding nodes may re-broadcast a message immediately upon receiving it).

## A.5 Performance Evaluation

We performed a simulation study to evaluate our centralized algorithm (EWMA) and its distributed version.

We compared the centralized version of our algorithm (EWMA) with BIP and MST algorithms. We performed simulations using networks of four different sizes: 10, 30, 50 and 100 nodes. The nodes in the networks are distributed according to a spatial Poisson distribution over the same deployment region. Thus, the higher the number of nodes, the higher the network density. The source node for each simulation is chosen randomly from the overall set of nodes. The maximum transmission range is chosen such that each node can reach all other nodes in the network. The transmission power used by a node in transmission ( $d^\alpha$ ) depends on the reached distance  $d$ , where the propagation loss exponent  $\alpha$  is varied. Similarly to Wieselthier et al. in [108], we ran 100 simulations for each simulation setup consisting of a network of a specified size, a propagation loss exponent  $\alpha$ , and an algorithm.

The performance metric used is the total power of the broadcast tree. Here we use the idea of the *normalized tree power* [108]. Let  $p_i(m)$  denote the total power of the broadcast tree for a network instance  $m$ , generated by algorithm  $i = \{EWMA, BIP, MST\}$ . Let  $p_0$  be the power of the lowest-power broadcast tree among the set of algorithms performed and all network instances (100 in our case). Then the normalized tree power associated with algorithm  $i$  and network instance  $m$  is defined as follows:  $p'_i(m) = \frac{p_i(m)}{p_0}$ .

Let us first consider the performance of the algorithms shown in Figure A.4. We can see the average normalized tree power (shown on the vertical axis) achieved by the algorithms on networks of different sizes (the horizontal axis) for (a)  $\alpha = 2$ , (b)  $\alpha = 4$ . To estimate the average power, we used an interval estimate with a confidence interval of 95%. The figure shows that the solutions for



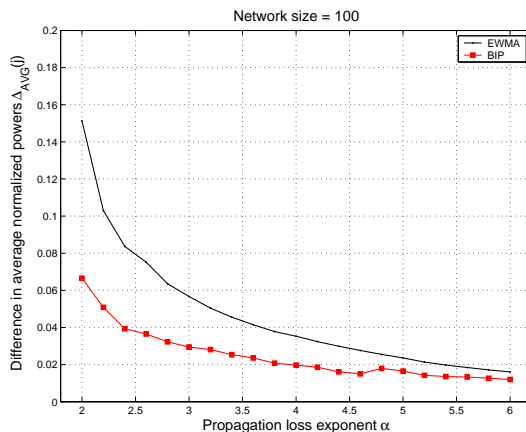


Figure A.5: Normalized tree power as a function of the propagation loss exponent  $\alpha$

the broadcast tree obtained by EWMA have, on the average, lower costs than the solutions of BIP and MST. (This is also true for  $\alpha = 3$ , which is not shown in the figure). However, we notice that for the propagation loss exponent of  $\alpha = 4$ , the confidence intervals of the algorithms overlap for certain cases, which means that the solutions obtained by the algorithms are not significantly different. Thus the figure also reveals that the difference in performance decreases as the propagation loss exponent increases. This is better seen in Figure A.5, where the difference in the average normalized tree powers between EWMA (BIP) and MST algorithms ( $\Delta_{AVG}(j) = AVG(p'_{MST}(m)) - AVG(p'_j(m))$ ,  $j = \{BIP, EWMA\}$ ) is shown for different values of the propagation loss exponent (the horizontal axis). Notice here that the larger the difference  $\Delta_{AVG}(j)$ , the lower the cost of the broadcast tree. The main reason for such behaviour is that by increasing the propagation loss exponent, the cost of using longer links increases as well. Consequently, EWMA and BIP select their transmitting nodes to transmit using lower power levels, which is typical for the transmitting nodes of MST. Hence, in a sense, EWMA and BIP's broadcast trees *converge* to the MST tree when  $\alpha$  increases. This indicates that in scenarios where  $\alpha$  takes higher values, MST performs quite well.

We also conducted a simulation study of the distributed algorithm presented in Section A.4.3. The performance metric used here is the same as in the case of the centralized algorithm and is based again on the normalized tree power. However, here we do not consider the cost of building a broadcast tree, but only the cost of the final tree produced by the distributed algorithm. The performance of the distributed algorithm is compared to that of the centralized algorithms, and is shown in Figure A.6. We can see that broadcast trees produced by distributed EWMA have, on the average, lower costs than those obtained by the centralized BIP and MST. Also, we can see that distributed EWMA performs almost as well as its centralized counterpart. Note that the results for the centralized algorithms differ between Figure A.4(a) and Figure A.6. This is because here we run another set of simulations for all the algorithms, and for each network the source node is chosen at random.

Based on our simulation results, we conclude that EWMA utilizes the wireless multicast advantage property at least as well as BIP. The main problem with BIP is that it is not easy to distribute. On the other hand, we showed here that EWMA can be easily distributed.

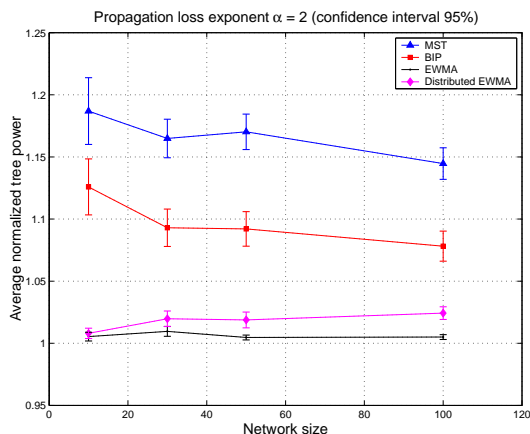


Figure A.6: Normalized tree power comparison (Distributed EWMA;  $\alpha = 2$ )

## A.6 Related Work

Minimizing the energy consumption of all-wireless networks has received significant attention over the last few years [28, 91, 47, 71, 114, 72, 95, 54]. We were inspired by the exciting results related to the problem of minimum-energy broadcasting in all-wireless networks [97, 105, 36, 70, 78, 81], and in particular by the work of Wieselthier et al. [108, 109]. In this work they introduce the node-based multicast model for wireless networks upon which they have built several broadcast and multicast heuristics. One of the most notable contributions of their work is the Broadcast Incremental Power (BIP) algorithm. The main objective of BIP is to construct a minimum-energy broadcast tree rooted at the source node. It constructs the tree by first determining the node that the source can reach with a minimum expenditure of power. BIP constructs a tree that initially contains a single node; it then determines which uncovered node can be added to the tree at a *minimum additional* cost. At each iteration of BIP, the nodes that have already covered some node can further increase their transmission power to reach some other yet uncovered node. BIP is similar to Prim’s algorithm [31] for the formation of minimum spanning trees, but with the difference that weights, with BIP, are dynamically updated at each step.

In [105] Wan et al. provide the first analytical results for the minimum energy broadcast problem. By exploring geometric structures of an Euclidean minimum spanning tree (MST), they prove that the approximation ratio of MST is between 6 and 12, and that the approximation ratio of BIP is between  $\frac{13}{6}$  and 12. They also found that for some instances, BIP fails to use the broadcast nature of the wireless channel. This happens because BIP adds only one node at each iteration, the one that can be added at a minimum additional cost. Thus BIP, although centralized, does not use all the available information about the network. As a result, it may construct a broadcast tree that coincides with the shortest path tree of a network graph, where the broadcast nature of the media is completely ignored. A possible approach to alleviate with this problem is to add to the tree more than one node at each iteration, and not necessarily at a minimum additional cost. But, in this case, there must be another criterion for the selection of nodes. Another difficulty with BIP is that distributing it is not obvious and according to the authors of BIP and of [105], the development of distributed algorithms is the major challenge for a minimum energy broadcast problem. However, Wan et al. [105] and Wieselthier et al. [108] do not really address this challenge. In Section A.4.2 we

present a heuristic algorithm that achieves the same approximation ratio as BIP for the geometric case, and yet is easy to distribute.

Li et al., in another closely related work [70], also recognize weaknesses of BIP and propose another centralized heuristic to tackle the broadcasting problem. However, they do not consider the issue of developing a distributed algorithm for a minimum energy broadcast. Li et al. [70] also sketch a proof of the NP-hardness of a general version of the minimum energy broadcast.

A proof of NP-hardness of the minimum energy broadcast problem in metric space has been proposed by Egecioglu et al. [36]. However, in their interpretation of the minimum energy broadcast problem, they restrict nodes transmission ranges a set of *integers*, which captures very few instances of the problem in metric space.

In [73] Liang provide a proof of NP-completeness of the minimum-energy broadcast problem, as well as an approximation algorithm for the problem in general setting, which delivers a feasible solution of cost  $O(\log^3 n)$  times the optimum. The time complexity of the proposed algorithm is  $O(kn^2 \log n)$ , where  $k$  is the total number of power levels at each node. However, [73] does not provide an answer to the question: Whether there is an approximation algorithm for the problem with a constant performance ratio. We do provide an answer in Section A.3 (Theorem 16).

Very recently, it was brought to our attention that more researchers are also studying the problem of minimum-energy broadcasting in all-wireless networks [29, 37].

## A.7 Summary

We have provided novel contributions on several relevant aspects of power-efficient broadcasting in all-wireless networks. First, we studied the complexity of the problem: we discussed two configurations, represented each by a specific graph - a general graph and a graph in Euclidean space (geometric case). For both, we showed that the problem is NP-complete. Furthermore, we showed that the general version cannot be approximate better than  $O(\log N)$ .

Second, we elaborated an approximation algorithm for the general version that achieves approximation ratio of  $18 \log N$ . Then we elaborated a new algorithm called Embedded Wireless Multicast Advantage (EWMA) that compares well with the existing proposals. Finally, we explained how centralized EWMA can be converted to a distributed algorithm, which is almost as energy-efficient as its centralized counterpart.

In future work we intend to study how to cope with the mobility of the nodes and study the minimum-energy multicast problem.



# Bibliography

- [1] *BTnodes*. <http://www.btnode.ethz.ch/>.
- [2] Mica sensor platform. <http://www.xbow.com>.
- [3] *Network Simulator (NS)*. <http://www.isi.edu/nsnam/ns/>.
- [4] RFC 2289 - A One-Time Password System. <http://www.ietf.org/rfc/rfc2289.txt?number=2289>.
- [5] RFC 2409 - The Internet Key Exchange (IKE). <http://www.ietf.org/rfc/rfc2409.txt?number=2409>.
- [6] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1999. The Institute of Electrical and Electronics Engineers, New York, 1999.
- [7] Specification of the Bluetooth System (Core). Version 1.1. <http://www.bluetooth.org>, 2001.
- [8] W. Aiello, S. M. Bellovin, M. Blaze, R. Canettia, J. Ioannidis, A. D. Keromytis, and O. Reinhold. Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols. In *Proceedings of ACM Computer and Communications Security (CCS) Conference*, pages 48–58, Washington, DC, 2000.
- [9] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communication Magazine*, 40(8):102–114, 2002.
- [10] T. Alpcan, T. Basar, R. Srikant, and E. Altman. CDMA uplink power control as a noncooperative game. In *Proceedings of 40th IEEE Conference on Decision and Control*, 2001.
- [11] B. Alpern and F. Schneider. Key exchange using Keyless Cryptography. *Information processing letters*, 16(2):79–82, 1983.
- [12] E. Altman, R. El Azouzi, and T. Jiménez. Slotted Aloha as a stochastic game with partial information. In *Proceedings of WiOpt*, 2002.
- [13] N. Asokan and P. Ginzboorg. Key Agreement in Ad-hoc Networks. *Computer Communications*, 23(17):1627–1637, November 2000.
- [14] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, 2002.
- [15] J. Bellardo and S. Savage. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In *Proceedings of the 12th USENIX Security Symposium*, 2002.

- [16] M. Bellare, R. Canetti, and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In *Proceedings of the 30th Annual Symposium on the Theory of Computing*. ACM, 1998.
- [17] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology – Crypto 93 Proceedings*. Lecture Notes in Computer Science Vol. 773, Springer-Verlag, 1993.
- [18] J. M. Berger. A Note on Error Detecting Codes for Asymmetric Channel. *Information and Control*, 4:68–73, 1961.
- [19] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal of Selected Areas in Communications*, 18, 2000.
- [20] M. Blaum and H. van Tilborg. On  $t$ -Error Correcting/All Unidirectional Error Detecting Codes. *IEEE Transactions on Computers*, pages 1493–1501, 1989.
- [21] J. M. Borden. Optimal Asymmetric Error Detecting Codes. *Information and Control*, 53:66–73, 1982.
- [22] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [23] S. Brands and D. Chaum. Distance-bounding protocols. In *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359. Springer-Verlag New York, Inc., 1994.
- [24] M. Cagalj and J.-P. Hubaux. Key agreement over a radio link. Technical Report IC/2004/16, EPFL-DI-ICA, January 2004.
- [25] M. Cagalj, J.-P. Hubaux, and C. Enz. Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networks (MOBICOM 2002)*, Atlanta, Georgia, September 2002.
- [26] S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 5(1), January 2006.
- [27] C. Castelluccia and P. Mutaf. Shake Them Up! A movement-based pairing protocol for CPU-constrained devices. In *Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys)*, 2005.
- [28] J.H. Chang and L. Tassiulas. "Energy Conserving Routing in Wireless Ad-hoc Networks". In *Proceedings of IEEE INFOCOM 2000*. ACM, 2000.
- [29] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocco. On the complexity of computing minimum energy consumption broadcast subgraphs. In *Proceedings of 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, 2001.
- [30] LAN/MAN Standards Committee. *ANSI/IEEE Std 802.11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, 1999.
- [31] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms. Second Edition*. The MIT Press, 2001.

- [32] M. Corner and B. Noble. Protecting applications with transient authentication. In *First ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys'03)*, San Francisco, CA, May 2003.
- [33] Steve Dohrmann and Carl Ellison. Public-key Support for Collaborative Groups. In *Proceedings of the 1st Annual PKI Research Workshop*, 2002.
- [34] A. D.Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, 2002.
- [35] L. Eschenauer and V. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2002.
- [36] Ö. Eğecioğlu and T.F. Gonzalez. Minimum-energy broadcast in simple graphs with limited node power. In *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001)*, pages 334–338, Anaheim, CA, August 2001.
- [37] A. Faragó and V.R. Syrotiuk. Algorithmic problems in power controlled ad hoc networks. In *Proceedings of the 14th International Conference on Parallel and Distributed Computing Systems (PDCS 2001)*, Dalas, Texas, August 2001.
- [38] R.J. Fontana, E. Richley, and J. Barney. Commercialization of an Ultra Wideband Precision Asset Location System. In *IEEE Conference on Ultra Wideband Systems and Technologies*, November 2003.
- [39] F. Forgó, J. Szép, and F. Szidarovszky. *Introduction to the Theory of Games: Concepts, Methods, Applications*. Kluwer Academic Publishers, 1999.
- [40] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, 1991.
- [41] R.G. Gallager, P.A. Humblet, and P.M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1983.
- [42] M.R. Garey and D.S. Johnson. *Computers and intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [43] C. Gehrman, C.J. Mitchell, and K. Nyberg. Manual Authentication for Wireless Devices, January 2004. RSA Cryptobytes, Vol. 7, No. 1.
- [44] C. Gehrman and K. Nyberg. Enhancements to Bluetooth Baseband Security. In *Proceedings of Nordsec*, Copenhagen, Denmark, November 2001.
- [45] V. Gupta, S. Krishnamurthy, and M. Faloutsos. Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks. In *Proceedings of MILCOM*, Anaheim, CA, 2002.
- [46] S. Han, R. Rengaswamy, R. Shea, E. Kohler, and M. Srivastava. A Dynamic Operating System for Sensor Nodes. In *Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys)*, 2005.
- [47] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, Maui, Hawaii, January 2000.

- [48] Dorit S. Hochbaum. *Approximation Algorithms for NP-hard problems*. PWS Publishing Company, 1997.
- [49] Y. Hu, A. Perrig, and D. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2003.
- [50] J.-H. Hoepman. The Ephemeral Pairing Problem. In *Financial Cryptography, FC04*, Key West, Florida, USA, February 2004.
- [51] M. Jakobsson. Payments and Diffie-Hellman key exchange (presentation slides). Private communication with M. Jakobsson.
- [52] M. Jakobsson and S. Wetzel. Security Weaknesses in Bluetooth. In *Progress in Cryptography - CT-RSA 2001*. Lecture Notes in Computer Science 2020, Springer-Verlag, February 2001.
- [53] Y. Jin and G. Kesidis. Equilibria of a noncooperative game for heterogeneous users of an ALOHA network. *IEEE Comm. Letters*, 6, 2002.
- [54] E.-S. Jung and N. Vaidya. “An Energy Efficient MAC Protocol for Wireless LANs”. In *IEEE INFOCOM 2002*, New York, USA, June 2002.
- [55] W. Kaiser, G. Pottie, M. Srivastava, G.S. Sukhatme, J. Villasenor, and D. Estrin. Networked Infomechanical Systems (NIMS) for Ambient Intelligence. In *Ambient Intelligence, Springer-Verlag*, 2004.
- [56] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier’s AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.
- [57] C. E. Koksal, H. Kassab, and H. Balakrishnan. An analysis of short-term fairness in wireless media access protocols. In *Proceedings of ACM Sigmetrics*, 2000.
- [58] J. Konorski. Multiple access in ad hoc wireless LANs with noncooperative stations. In *NETWORKING*, volume 2345 of LNCS, Springer, 2002.
- [59] Y. A. Korilis, A. A. Lazar, and A. Orda. Architecting noncooperative networks. *IEEE Journal of Selected Areas in Communications*, 13(7):1241–1251, 1995.
- [60] B. Korte and J. Vygen. *Combinatorial Optimization. Theory and Algorithms*. Springer-Verlag, 2000.
- [61] H. Krawczyk. SIGMA. <http://www.ee.technion.ac.il/hugo/sigma.html>.
- [62] D. Kügler. Man in the Middle Attacks on Bluetooth. In *Financial Cryptography*, Long Beach, 2003. Lecture Notes in Computer Science, Springer-Verlag.
- [63] P. Kyasanur and N. Vaidya. Detection and handling of MAC layer misbehavior in wireless networks. In *Dependable Systems and Networks*, 2003.
- [64] Kyasanur, P. and Vaidya, N. Detection and handling of MAC layer misbehavior in wireless networks. In *Dependable Systems and Networks*, June 2003.



- [65] R. J. La and V. Anantharam. Optimal routing control: repeated game approach. *IEEE Transactions on Automatic Control*, March 2002.
- [66] J.-O. Larsson and M. Jakobsson. SHAKE. Private communication with M. Jakobsson.
- [67] E. L. Leiss. Data Integrity on Digital Optical Discs. *IEEE Transactions on Computers*, 33:818–827, 1984.
- [68] A. Lenstra and E. R. Verheul. Selecting Cryptographic Key Sizes. 1999.
- [69] M. Leopold, M.B. Dydensborg, and P. Bonnet. Bluetooth and Sensor Networks: A Reality Check. In *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*.
- [70] F. Li and I. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, Tampa, Florida, November 2001.
- [71] L. Li, V. Bahl, Y.M. Wang, and R. Wattenhofer. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of IEEE INFOCOM 2001*, April 2001.
- [72] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MOBICOM 2001)*, Rome, Italy, July 2001.
- [73] W. Liang. Constructing Minimum-Energy Broadcast Trees In Wireless Ad Hoc Networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, Lausanne, Switzerland, June 2002.
- [74] R.-F. Liao, Rita H. Wouhaybi, and A.T. Campbell. Incentive engineering in wireless LAN based access networks. In *Proceedings of 10th International Conference on Network Protocols (ICNP 2002)*, 2002.
- [75] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, May 1982.
- [76] A. B. MacKenzie and S. B. Wicker. Stability of multipacket slotted Aloha with selfish users and perfect information. In *Proceedings of IEEE INFOCOM*, 2003.
- [77] D.P. Maher. United States Patent (No. 5,450,493): Secure communication method and apparatus. <http://www.uspto.gov>, 1993.
- [78] K. Makki, N. Pissinou, and O. Frieder. Efficient Solutions to Multicast Routing in Communication Networks. *ACM Mobile Networks and Applications (MONET)*, 1(2):221–232, 1996.
- [79] Wenbo Mao. *Modern Cryptography, Theory & Practice*. Prentice Hall PTR, 2004.
- [80] R. C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In C. Pomerance, editor, *Lecture Notes in Computer Science (CRYPTO'87)*, volume 293, pages 369–378. Springer-Verlag, 1988.
- [81] M. Nagy and S. Singh. Multicast Scheduling Algorithms in Mobile Networks. *Cluster Computing*, 1(2):177–185, 1998.

- [82] G. Noubir and G. Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):29–30, 2003.
- [83] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multi-user communication networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.
- [84] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.
- [85] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(Summer), 2002.
- [86] A. Perrig and D. Song. Hash Visualization: A New Technique to Improve Real-World Security. In *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pages 131–138, July 1999.
- [87] D. Plummer. An Ethernet Address Resolution Protocol, November 1982. IETF Standards Track RFC 826.
- [88] John G. Proakis and Masoud Salehi. *Communication Systems Engineering – Second Edition*. Prentice Hall, 2002.
- [89] Theodore S. Rappaport. *Wireless communications: Principles and Practice*. Prentice Hall, 1996.
- [90] M. Raya, J.-P. Hubaux, and I. Aad. DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots. In *Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2004.
- [91] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.
- [92] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman. Efficient power control via pricing in wireless data networks. *IEEE Transactions on Communications*, 50(2):291–303, February 2002.
- [93] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, pages 1–10, September 2003.
- [94] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag Telos, 1999.
- [95] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.
- [96] G. Sharma and R.R. Mazumdar. Hybrid Sensor Networks: A Small World. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, Illinois, USA, 2005.
- [97] S. Singh, C. Raghavendra, and J. Stepanek. Power-aware broadcasting in mobile ad hoc networks. In *Proceedings of IEEE PIMRC'99*, Osaka, Japan, September 1999.
- [98] Herbert Solomon. *Geometric Probability*. SIAM, 1978.

- [99] D. Song. dsniff. <http://naughty.monkey.org/~dugsong/dsniff/>.
- [100] D. Song. Passwords Found on a Wireless Network. In *USENIX Technical Conference WIP*, June 2000.
- [101] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proceedings of the 7th International Workshop on Security Protocols*, 1999.
- [102] Frank Stajano. *Security for Ubiquitous Computing*. John Wiley & Sons, Ltd., 2002.
- [103] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- [104] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, pages 644–654, November 1976.
- [105] P.-J. Wan, G. Calinescu, and O. Frieder X.-Y. Li. Minimum-energy broadcast routing in static ad hoc wireless networks. In *IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.
- [106] B. Waters and E. Felten. Proving the Location of Tamper-Resistant Devices. Technical report, Princeton University.
- [107] IEEE 802.11 WG. *ANSI/IEEE Std 802.11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS) IEEE 802.11/D2.0*. IEEE, 2001.
- [108] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *IEEE INFOCOM 2000*, pages 586–594, Tel Aviv, Israel, 2000.
- [109] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *ACM Mobile Networks and Applications (MONET)*, 6(3):251–263, June 2001.
- [110] A.D. Wood, J.A., Stankovic, and S.H. Son. JAM: A Jammed-Area Mapping Service for Sensor Networks. In *Real-Time Systems Symposium (RTSS)*, Cancun, Mexico, 2003.
- [111] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, Illinois, USA, 2005.
- [112] W. Xu, T. Wood, W. Trappe, and Y. Zhang. Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2004.
- [113] H. Yaïche, R. Mazumdar, and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking*, 8(5):667–678, 2000.
- [114] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MOBICOM 2001)*, July 2001.

# Index

- $G_{\text{CSMA/CA}}^\infty$ , repeated CSMA/CA game, 23
  - $p_i$  penalty function, 23
  - adaptive strategy, 31
  - coordinator, 27
  - detection mechanism, 30
  - equilibrium coordination algorithm, 27
  - full dimensionality, 24
  - penalizing mechanism, 24
  - selective jamming, 25
  - Subgame Perfect Nash Equilibrium, 24
- $ns$ -2, network simulator, 14
- authenticated link model (AM), 74
- authentication through presence, 90, 101
- Bianchi's model
  - $\tau$ , channel access probability, 12
- Commitment schemes, 70
- CSMA/CA cooperative game
  - $r^\circ$ , disagreement point, 19
  - cooperative players, 6
  - Pareto optimal point, 22
- CSMA/CA noncooperative game, 12
  - $W_i$ , contention window size, 11
  - $\hat{G}_{\text{CSMA/CA}}$ , an approximate game, 17
  - $\mathcal{I}$ , set of cheaters, 11
  - $W_\infty$ , player does not transmit, 11
  - $r_i^{(c)}$ , throughput of cheater  $i$ , 13
  - $r_i$ , average throughput, 11
  - Nash equilibrium, 16
  - Pareto optimal Nash equilibrium, 16
  - selfish user, 5
  - well-behaved nodes, 12
- Game theory
  - $S_i$ , pure strategy set, 7
  - $\sigma_i$ , mixed strategy, 7
  - $u_i$ , payoff function, 7
  - $B_i$ , best-response function, 8
- essential game, 10
- essential Nash equilibrium, 10
- Nash equilibrium, 7
- noncooperative games, 7
- normal form, 7
- strictly dominated strategy, 7
- weakly dominated strategy, 7
- integrity codes ( $I$ -codes), 89
  - $P_0$ , threshold power level, 94
  - $T_s$ , symbol period, 94
  - bit flipping, 91
  - codes with fixed Hamming weight, 92
  - complementary encoding, 92
  - definition, 92
  - distance shift, 107
  - implementation, 99
  - incongruous delimiter, 97
  - overshadowing attack, 91
  - phase shift, 106
  - signal spreading, 95
  - synchronization, 96
  - time shift, 107
- integrity region, 83
- Key agreement
  - DH-DB, Diffie-Hellman key agreement based on Distance Bounding, 81
  - DH-IC, Diffie-Hellman key agreement protocol based on  $I$ -codes, 104
  - DH-SC, Diffie-Hellman key agreement protocol based on String Comparison, 77
  - Diffie-Hellman, 68
  - distance bounding, 82
  - distance-bounding, 68
  - low-bandwidth authentication channel, 69
  - time-invariant users' involvement, 71
  - time-variant users' involvement, 71

- user-friendly, 68
- matching conversations, 78
- message transfer (MT) authenticator, 68, 70
- message transfer (MT) authenticator, 72
- minimum broadcast cover, 117, 119
  - geometric version, 120
- minimum-energy broadcast problem, 115
  - $O(\log N)$ -approximation algorithm, 120
  - complexity, 116
  - EWMA, Embedded Wireless Multicast Advantage, 124
- minimum-energy broadcasting
  - trees, 115
  - WMA, Wireless Multicast Advantage, 116
- Nash Bargaining Framework
  - $B$ , bargaining problem, 8
  - $IIA$ , independence of irrelevant alternatives, 9
  - $IUO$ , independence of utility origins, 9
  - $IUU$ , independence of utility units, 9
  - $NBS$ , Nash Bargaining Solution, 9
  - $P$ , Pareto property, 9
  - $S$ , symmetry property, 9
  - $U$ , feasible payoffs set, 8
  - $u^\circ$ , disagreement point, 8
- NP-completeness, 117
- on-off keying, 89, 94
- peer-to-peer, 67
- peer-to-peer security, 67
- proactive sensor networks, 39
- reactive sensor network, 39
- reflection attack, 76
- signal anti-blocking, 89, 105
- signal energy, 106, 110
- signal spreading, 108
- tragedy of the commons, 16
- un-authenticated links model (UM), 74
- Wormholes
  - $K_0$ , critical number of connected pairs, 47
  - $R_j$ , jamming range, 41
  - $R_j$ , jamming region, 41
  - $R_s$ , exposure region, 41
  - $R_t$ , transmission range, 42
  - $P$ [at least one wormhole $|(x_A, y_A)$ ], probability that at least one wormhole exists, 54
  - $P$ [at least one wormhole $|(x_A, y_A)$ ], probability that at least one wormhole exists, 43
  - $\rho_j$ , jamming ratio, 57, 58
  - coordinated frequency hopping pairs, 50
  - coverage paradox, 37
  - disk line picking model, 54
  - event-masking attacks, 37
  - exposure region, 38
  - FH enabled nodes, 50
  - hybrid sensor network, 37, 40
  - jamming region, 38
  - opportunistic pairing process, 50
  - probabilistic, 40
  - probabilistic wormholes, 37
  - uncoordinated channel-hopping, 56
  - wired pairs, 41



# Curriculum Vitae

MARIO ČAGALJ

Born in Imotski, Croatia on December 10, 1975  
Nationality: Croatian

## EDUCATION

- EPFL, **PhD in Communication Systems**, *January 2006*  
Thesis title: *Thwarting Selfish and Malicious Behavior in Wireless Networks*  
Thesis advisor: *Prof. Jean-Pierre Hubaux*
- EPFL, **Doctoral School in Communication Systems**, *October 2000 - July 2001*
- University of Split, Croatia, **BSc in Electrical Engineering**, *September 1994 - December 1998*

## PUBLICATIONS

- M. Felegyhazi, M. Čagalj and J.-P. Hubaux “Multi-Radio Channel Allocation in Competitive Wireless Networks”. *International Workshop on Incentive-Based Computing (IBC'06)*, Lisboa, Portugal, 2006.
- M. Čagalj, S. Čapkun, R. Rengaswamy, I. Tsigkogiannis, M. Srivastava and J.-P. Hubaux. “Integrity (I) codes: Message Integrity Protection and Authentication over Insecure Channels”. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, USA, 2006.
- M. Čagalj, S. Čapkun and J.-P. Hubaux. “Key agreement in peer-to-peer wireless networks”. *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, vol. 94, no. 2, pp. 467–478, February, 2006.
- S. Čapkun, M. Čagalj and M. Srivastava “Securing Localization With Hidden and Mobile Base Stations”. In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, 2006.
- M. Čagalj, J.-P. Hubaux and C. Enz. “Energy-Efficient Broadcasting in All-Wireless Networks”. *Wireless Networks (WINET)*, 11:177–188, 2005.
- M. Čagalj, S. Ganeriwal, I. Aad and J.-P. Hubaux. “On Selfish Behavior in CSMA/CA Networks”. In *Proceedings of IEEE INFOCOM '05*, Miami, Florida, March 2005.
- M. Čagalj, J.-P. Hubaux and C. Enz. “Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues”. In *Proceedings of MOBICOM '02*, Atlanta, Georgia, September 2002.

## Submitted for publishing

- M. Čagalj, S. Čapkun and J.-P. Hubaux. “Wormhole-Based Anti-Jamming Techniques in Sensor Networks”, 2006.

- S. Čapkun and M. Čagalj. “Integrity Regions: Authentication Through Presence in Wireless Networks”, 2006.

## **SUPERVISED PROJECTS**

- March 2005 – June 2005: Alexandre Kozma, “Resource Allocation in Competitive Wireless Networks”. *Semester project.*
- June 2004 – February 2005: Cédric Renouard, “Rational DoS Attacks Against IEEE 802.11b”. *Diploma project.*
- November 2003 – February 2004: Jim Pugh. “Cyclex: A New Cheat-Proof Wireless MAC Protocol”. *Doctoral course project.*
- October 2003 – February 2004: Velik Bellemin, “Real-time Implementation of Backoff Cheating Detection”. *Semester project.*
- February 2003 – June 2003: Alexander Zurkinden, “Performance Evaluation of AODV Routing Protocol: Real-Life Measurements”. *Semester project.*

## **TEACHING**

- Teaching assistant, Mobile Networks, EPFL, 2004–present
- Teaching assistant, Self-Organized Wireless and Sensor Networks, EPFL, 2004–present

## **PROFESSIONAL ACTIVITIES**

- Reviewer for scientific journals: IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, Proceedings of the IEEE, IEEE Communication Magazine, Ad Hoc Networks
- Reviewer for scientific conferences and workshops: ACM MobiCom, IEEE InfoCom, ACM MobiHoc, ACM WiSe, ACM SASN, ESAS, IEEE SecureComm

## **PROFESSIONAL EXPERIENCE**

- March 2000 – October 2000: Java software developer in SIEMENS d.d., Zagreb, Croatia.
- October 1999 – March 2000: System manager in SEM Maritime company, Split, Croatia.

## **LANGUAGES**

English, Croatian