ABSTRACT

Title of dissertation: DESIGN AND EVALUATION OF DECISION MAKING ALGORITHMS FOR INFORMATION SECURITY

Alvaro A. Cárdenas, Doctor of Philosophy, 2006

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

The evaluation and learning of classifiers is of particular importance in several computer security applications such as intrusion detection systems (IDSs), spam filters, and watermarking of documents for fingerprinting or traitor tracing. There are however relevant considerations that are sometimes ignored by researchers that apply machine learning techniques for security related problems. In this work we identify and work on two problems that seem prevalent in security-related applications. The first problem is the usually large class imbalance between normal events and attack events. We address this problem with a unifying view of different proposed metrics, and with the introduction of Bayesian Receiver Operating Characteristic (B-ROC) curves. The second problem to consider is the fact that the classifier or learning rule will be deployed in an adversarial environment. This implies that good performance on average might not be a good performance measure, but rather we look for good performance under the worst type of adversarial attacks. We work on a general methodology that we apply for the design and evaluation of IDSs and Watermarking applications.

DESIGN AND EVALUATION OF DECISION MAKING ALGORITHMS FOR INFORMATION SECURITY

by

Alvaro A. Cárdenas

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Virgil D. Gligor
Professor Bruce Jacob
Professor Carlos Berenstein
Professor Nuno Martins

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

Look if I can raise the money fast, can I set up my own lab?
-The Smithsonian Institution. Gore Vidal

The algorithms used to ensure several information security goals, such as authentication, integrity and secrecy, have often been designed and analyzed with the help of formal mathematical models. One of the most successful examples is the use of theoretical cryptography for encryption, integrity and authentication. By assuming that some basic primitives hold, (such as the existence of one-way functions), some cryptographic algorithms can formally be proven secure.

Information security models have however theoretical limits, since it cannot always be proven that an algorithm satisfies (or not) certain security conditions. For example, as formalized by Harrison, Ruzzo, and Ullman, the access matrix model is undecidable and Rices theorem implies that static analysis problems are also undecidable. Because of similar results such as the undecidability of detecting computer viruses, there is reason to believe that several intrusion detection problems are also undecidable.

Undecidability is not the only problem for being able to formally characterize the security properties of an algorithm. Not only are some problems intractable, or undecidable, but also there are certain inherent uncertainties in several security related problems such as biometrics, data hiding (watermarking), fraud detection and spam filters that are impossible to factor out.

All algorithms trying to solve these problems will make a non-negligible amount of decision errors. These errors occur due to approximations and can consequently be formulated as a tradeoff between the costs of operation (e.g., the necessary resources for their operation, such as the number of false alarms) and the correctness of their output (e.g., the security level achieved, or the probability of detection). It is however very difficult in practice to assess both: the real costs of these security solutions and their actual security guarantees. Most of the research therefore relies on ad hoc solutions and heuristics that cannot be shared between security fields trying to address these hard problems.

In this work we try to address the problem of providing a necessary framework in order to reason about the security and the costs of an algorithm for solving problems where the decision between two hypotheses cannot be made without errors.

Our framework is composed of two main parts.

Evaluation Metrics: We introduce in a unified framework several metrics that have been previously proposed in the literature. We give intuitive interpretations of them and provide new metrics that address two of the main problems for decision algorithms. First, the large class imbalance between the two hypotheses, and second, the uncertainty of several parameters, including costs and the class imbalance severity.

Security Model: In order to reason formally about the security level of a decision algorithm, we need to introduce a formal model for an adversary and the system being evaluated. We therefore clearly define the feasible design space, and the properties our algorithm

need to satisfy by an evaluation metric. Then we model the considered adversary class by clearly defining the information available to the adversary, the capabilities of the adversary and its goal. Finally, since the easiest way to break the security of a system is to step outside the box, i.e., to break the rules under which the algorithm was evaluated, we clearly identify the assumptions made in our model. These assumptions then play an important part in the evaluation of the algorithm. Our tools to solve and analyze this models are based on robust detection theory and game theory, and the aim is always the same, minimize the advantage of the adversary (the advantage of the adversary is defined according to the metric used), for any possible adversary in a given adversary class.

We apply our model to different problems in intrusion detection, selfish packet forwarding in ad hoc networks, selfish MAC layer misbehavior in wireless access points and watermarking algorithms. Our results show that formally modeling these problems and obtaining the least favorable attack distribution can lead in the best cases to show how there is merit in our framework, by outperforming previously proposed heuristic solutions (analytically and by simulations), and in the worst cases they can be seen as pessimistic decisions based on risk aversion.

The following is the layout of this dissertation. In chapter 2 we view in a unified framework traditional metrics used to evaluate the performance of intrusion detection systems and introduce a new evaluation curve called the B-ROC curve. In chapter 3 we introduce the notion of security of a decision algorithm and define the adversarial models that are going to be used in the next chapters in order to design and evaluate decision making algorithms. In chapter 4 we compare the performance of two classification algorithms used for detecting MAC layer misbehavior in wireless networks. This chapter is a validation of our approach, since it shows how the design of classification algorithms using robust detection theory and formal adversarial modeling can outperform theoretically and empirically previously proposed algorithms based on heuristics. Finally in chapter 5 we apply our framework for the design and evaluation of data hiding algorithms.

Chapter 2
Performance Evaluation Under the Class Imbalance Problem

For there seem to be many empty alarms in war
-Nicomachean Ethics, Aristotle

I Overview

Classification accuracy in intrusion detection systems (IDSs) deals with such fundamental problems as how to compare two or more IDSs, how to evaluate the performance of an IDS, and how to determine the best configuration of the IDS. In an effort to analyze and solve these related problems, evaluation metrics such as the *Bayesian detection rate*, the *expected cost*, the *sensitivity* and the *intrusion detection capability* have been introduced. In this chapter, we study the advantages and disadvantages of each of these performance metrics and analyze them in a unified framework. Additionally, we introduce the Bayesian Receiver Operating Characteristic (B-ROC) curves as a new IDS performance tradeoff which combines in an intuitive way the variables that are more relevant to the intrusion detection evaluation problem.

II Introduction

Consider a company that, in an effort to improve its information technology security infrastructure, wants to purchase either intrusion detector 1 ($I\mathcal{DS}_1$) or intrusion detector 2 ($I\mathcal{DS}_2$). Furthermore, suppose that the algorithms used by each IDS are kept private and therefore the only way to determine the performance of each IDS (unless some reverse engineering is done [1]) is through empirical tests determining how many intrusions are detected by each scheme while providing an acceptable level of false alarms. Suppose these tests show with high confidence that $I\mathcal{DS}_1$ detects one-tenth more attacks than $I\mathcal{DS}_2$ but at the cost of producing one hundred times more false alarms. The company needs to decide based on these estimates, which IDS will provide the best return of investment for their needs and their operational environment.

This general problem is more concisely stated as the intrusion detection evaluation problem, and its solution usually depends on several factors. The most basic of these factors are the *false alarm rate* and the *detection rate*, and their tradeoff can be intuitively analyzed with the help of the *receiver operating characteristic* (ROC) curve [2, 3, 4, 5, 6]. However, as pointed out in [7, 8, 9], the information provided by the detection rate and the false alarm rate alone might not be enough to provide a good evaluation of the performance of an IDS. Therefore, the evaluation metrics need to consider the environment the IDS is going to operate in, such as the maintenance costs and the hostility of the operating environment (the likelihood of an attack). In an effort to provide such an evaluation method, several performance metrics such as the *Bayesian detection rate* [7], *expected cost* [8], *sensitivity* [10] and *intrusion detection capability* [9], have been proposed in the literature.

Yet despite the fact that each of these performance metrics makes their own contribution to the analysis of intrusion detection systems, they are rarely applied in the literature when

proposing a new IDS. It is our belief that the lack of widespread adoption of these metrics stems from two main reasons. Firstly, each metric is proposed in a different framework (e.g. information theory, decision theory, cryptography etc.) and in a seemingly ad hoc manner. Therefore an objective comparison between the metrics is very difficult.

The second reason is that the proposed metrics usually assume the knowledge of some uncertain parameters like the likelihood of an attack, or the costs of false alarms and missed detections. Moreover, these uncertain parameters can also change during the operation of an IDS. Therefore the evaluation of an IDS under some (wrongly) estimated parameters might not be of much value.

A. Our Contributions

In this chapter, we introduce a framework for the evaluation of IDSs in order to address the concerns raised in the previous section. First, we identify the intrusion detection evaluation problem as a multi-criteria optimization problem. This framework will let us compare several of the previously proposed metrics in a unified manner. To this end, we recall that there are in general two ways to solve a multi-criteria optimization problem. The first approach is to combine the criteria to be optimized in a single optimization problem. We then show how the intrusion detection capability, the expected cost and the sensitivity metrics all fall into this category. The second approach to solve a multi-criteria optimization problem is to evaluate a tradeoff curve. We show how the Bayesian rates and the ROC curve analysis are examples of this approach.

To address the uncertainty of the parameters assumed in each of the metrics, we then present a graphical approach that allows the comparison of the IDS metrics for a wide range of uncertain parameters. For the single optimization problem we show how the concept of *isolines* can capture in a single value (the slope of the isoline) the uncertainties like the likelihood of an attack and the operational costs of the IDS. For the tradeoff curve approach, we introduce a new tradeoff curve we call the Bayesian ROC (B-ROC). We believe the B-ROC curve combines in a single graph all the relevant (and intuitive) parameters that affect the practical performance of an IDS.

In an effort to make this evaluation framework accessible to other researchers and in order to complement our presentation, we started the development of a software application available at [11] to implement the graphical approach for the expected cost and our new B-ROC analysis curves. We hope this tool can grow to become a valuable resource for research in intrusion detection.

III Notation and Definitions

In this section we present the basic notation and definitions which we use throughout this document.

We assume that the input to an intrusion detection system is a feature-vector $\mathbf{x} \in \mathcal{X}$. The elements of $\mathbf{x}$ can include basic attributes like the duration of a connection, the protocol type, the service used etc. It can also include specific attributes selected with domain knowledge such as the number of failed logins, or if a superuser command was attempted. Examples of $\mathbf{x}$ used in intrusion detection are sequences of system calls [12], sequences of user commands [13], connection attempts to local hosts [14], proportion of accesses (in terms of TCP or UDP packets) to a given port of a machine over a fixed period of time [15] etc.

Let I denote whether a given instance $\mathbf{x}$ was generated by an intrusion (represented by $I = 1$ or simply I) or not (denoted as $I = 0$ or equivalently $\neg I$). Also let A denote whether the output of an IDS is an alarm (denoted by $A = 1$ or simply A) or not (denoted by $A = 0$, or equivalently $\neg A$). An IDS can then be defined as an algorithm $I\mathcal{DS}$ that receives a continuous data stream of computer event features $\mathbf{X} = \{\mathbf{x}[1], \mathbf{x}[2], \ldots, \}$ and classifies each input $\mathbf{x}[j]$ as being either a normal event or an attack i.e. $I\mathcal{DS} : \mathcal{X} \rightarrow \{A, \neg A\}$. In this chapter we do not address how the IDS is designed. Our focus will be on how to evaluate the performance of a given IDS.

Intrusion detection systems are commonly classified as either *misuse* detection schemes or *anomaly* detection schemes. Misuse detection systems use a number of attack signatures describing attacks; if an event feature $\mathbf{x}$ matches one of the signatures, an alarm is raised. Anomaly detection schemes on the other hand rely on profiles or models of the normal operation of the system. Deviations from these established models raise alarms.

The empirical results of a test for an IDS are usually recorded in terms of how many attacks were detected and how many false alarms were produced by the IDS, in a data set containing both normal data and attack data. The percentage of alarms out of the total number of normal events monitored is referred to as the *false alarm rate* (or the *probability of false alarm*), whereas the percentage of detected attacks out of the total attacks is called the *detection rate* (or *probability of detection*) of the IDS. In general we denote the probability of false alarm and the probability of detection (respectively) as:

$$P_{FA} \equiv \Pr[A = 1 | I = 0] \quad \text{and} \quad P_D \equiv \Pr[A = 1 | I = 1] \tag{2.1}$$

These empirical results are sometimes shown with the help of the ROC curve; a graph whose x-axis is the false alarm rate and whose y-axis is the detection rate. The graphs of misuse detection schemes generally correspond to a single point denoting the performance of the detector. Anomaly detection schemes on the other hand, usually have a monitored statistic which is compared to a threshold τ in order to determine if an alarm should be raised or not. Therefore their ROC curve is obtained as a parametric plot of the probability of false alarm (P_{FA}) versus the probability of detection (P_D) (with parameter τ) as in [2, 3, 4, 5, 6].

IV Evaluation Metrics

In this section we first introduce metrics that have been proposed in previous work. Then we discuss how we can use these metrics to evaluate the IDS by using two general approaches, that is the expected cost and the tradeoff approach. In the expected cost approach, we give intuition of the expected cost metric by relating all the uncertain parameters (such as the probability of an attack) to a single line that allows the IDS operator to easily find the optimal tradeoff. In the second approach, we identify the main parameters that affect the quality of the performance of the IDS. This will allow us to later introduce a new evaluation method that we believe better captures the effect of these parameters than all previously proposed methods.

A. Background Work

Expected Cost

In this section we present the expected cost of an IDS by combining some of the ideas originally presented in [8] and [16]. The expected cost is used as an evaluation method for

State of the system	Detector's report	
	No Alarm (A=0)	Alarm (A=1)
No Intrusion ($I=0$)	$C(0,0)$	$C(0,1)$
Intrusion ($I=1$)	$C(1,0)$	$C(1,1)$

Table 2.1: Costs of the IDS reports given the state of the system

IDSs in order to assess the investment of an IDS in a given IT security infrastructure. In addition to the rates of detection and false alarm, the expected cost of an IDS can also depend on the hostility of the environment, the IDS operational costs, and the expected damage done by security breaches.

A quantitative measure of the consequences of the output of the IDS to a given event, which can be an intrusion or not are the costs shown in Table 2.1. Here $C(0,1)$ corresponds to the cost of responding as though there was an intrusion when there is none, $C(1,0)$ corresponds to the cost of failing to respond to an intrusion, $C(1,1)$ is the cost of acting upon an intrusion when it is detected (which can be defined as a negative value and therefore be considered as a profit for using the IDS), and $C(0,0)$ is the cost of not reacting to a non-intrusion (which can also be defined as a profit, or simply left as zero.)

Adding costs to the different outcomes of the IDS is a way to generalize the usual tradeoff between the probability of false alarm and the probability of detection to a tradeoff between the *expected cost for a non-intrusion*

$$R(0,P_{FA}) \equiv C(0,0)(1-P_{FA}) + C(0,1)P_{FA}$$

and the *expected cost for an intrusion*

$$R(1,P_D) \equiv C(1,0)(1-P_D) + C(1,1)P_D$$

It is clear that if we only penalize errors of classification with unit costs (i.e. if $C(0,0) = C(1,1) = 0$ and $C(0,1) = C(1,0) = 1$) the expected cost for non-intrusion and the expected cost for intrusion become respectively, the false alarm rate and the detection rate.

The question of how to select the optimal tradeoff between the expected costs is still open. However, if we let the hostility of the environment be quantified by the *likelihood of an intrusion* $p \equiv \Pr[I=1]$ (also known as the *base-rate* [7]), we can average the expected non-intrusion and intrusion costs to give the overall *expected cost of the IDS*:

$$\mathbf{E}[C(I,A)] = R(0,P_{FA})(1-p) + R(1,P_D)p \tag{2.2}$$

It should be pointed out that $R()$ and $\mathbf{E}[C(I,A)]$ are also known as the *risk* and *Bayesian risk* functions (respectively) in Bayesian decision theory.

Given an IDS, the costs from Table 2.1 and the likelihood of an attack p, the problem now is to find the optimal tradeoff between P_D and P_{FA} in such a way that $\mathbf{E}[C(I,A)]$ is minimized.

The Intrusion Detection Capability

The main motivation for introducing the *intrusion detection capability* C_{ID} as an evaluation metric originates from the fact that the costs in Table 2.1 are chosen in a subjective way [9]. Therefore the authors propose the use of the intrusion detection capability as an objective

metric motivated by information theory:

$$C_{ID} = \frac{\mathbf{I}(I;A)}{\mathbf{H}(I)} \tag{2.3}$$

where **I** and **H** respectively denote the mutual information and the entropy [17]. The $\mathbf{H}(I)$ term in the denominator is a normalizing factor so that the value of C_{ID} will always be in the $[0,1]$ interval. The intuition behind this metric is that by fine tuning an IDS based on C_{ID} we are finding the operating point that minimizes the uncertainty of whether an arbitrary input event **x** was generated by an intrusion or not.

The main drawback of C_{ID} is that it obscures the intuition that is to be expected when evaluating the performance of an IDS. This is because the notion of reducing the uncertainty of an attack is difficult to quantify in practical values of interest such as false alarms or detection rates. Information theory has been very useful in communications because the entropy and mutual information can be linked to practical quantities, like the number of bits saved by compression (source coding) or the number of bits of redundancy required for reliable communications (channel coding). However it is not clear how these metrics can be related to quantities of interest for the operator of an IDS.

The Base-Rate Fallacy and Predictive Value Metrics

In [7] Axelsson pointed out that one of the causes for the large amount of false alarms that intrusion detectors generate is the enormous difference between the amount of normal events compared to the small amount of intrusion events. Intuitively, the base-rate fallacy states that because the likelihood of an attack is very small, even if an IDS fires an alarm, the likelihood of having an intrusion remains relatively small. Formally, when we compute the posterior probability of intrusion (a quantity known as the *Bayesian detection rate*, or the *positive predictive value* (PPV)) given that the IDS fired an alarm, we obtain:

$$\begin{aligned}
\text{PPV} &\equiv \Pr[I=1|A=1] \\
&= \frac{\Pr[A=1|I=1]\Pr[I=1]}{\Pr[A=1|I=1]\Pr[I=1]+\Pr[A=1|I=0]\Pr[I=0]} \\
&= \frac{P_D p}{(P_D - P_{FA})p + P_{FA}}
\end{aligned} \tag{2.4}$$

Therefore, if the rate of incidence of an attack is very small, for example on average only 1 out of 10^5 events is an attack ($p = 10^{-5}$), and if our detector has a probability of detection of one ($P_D = 1$) and a false alarm rate of 0.01 ($P_{FA} = 0.01$), then $\Pr[I=1|A=1] = 0.000999$. That is on average, of 1000 alarms, only one would be a real intrusion.

It is easy to demonstrate that the PPV value is maximized when the false alarm rate of our detector goes to zero, even if the detection rate also tends to zero! Therefore as mentioned in [7] we require a trade-off between the PPV value and the *negative predictive value* (NPV):

$$\text{NPV} \equiv \Pr[I=0|A=0] = \frac{(1-p)(1-P_{FA})}{p(1-P_D)+(1-p)(1-P_{FA})} \tag{2.5}$$

B. Discussion

The concept of finding the optimal tradeoff of the metrics used to evaluate an IDS is an instance of the more general problem of multi-criteria optimization. In this setting, we want to maximize (or minimize) two quantities that are related by a tradeoff, which can be done via two approaches. The first approach is to find a suitable way of combining these two metrics in a single objective function (such as the expected cost) to optimize. The second approach is to directly compare the two metrics via a trade-off curve.

We therefore classify the above defined metrics into two general approaches that will be explored in the rest of this chapter: the minimization of the expected cost and the tradeoff approach. We consider these two approaches as complimentary tools for the analysis of IDSs, each providing its own interpretation of the results.

Minimization of the Expected Cost

Let *ROC* denote the set of allowed (P_{FA}, P_D) pairs for an IDS. The expected cost approach will include any evaluation metric that can be expressed as

$$r^* = \min_{(P_{FA},P_D)\in ROC} \mathbf{E}[C(I,A)] \tag{2.6}$$

where r^* is the expected cost of the IDS. Given $I\mathcal{DS}_1$ with expected cost r_1^* and an $I\mathcal{DS}_2$ with expected cost r_2^*, we can say $I\mathcal{DS}_1$ is better than $I\mathcal{DS}_2$ for our operational environment if $r_1^* < r_2^*$.

We now show how C_{ID}, and the tradeoff between the PPV and NPV values can be expressed as an expected costs problems. For the C_{ID} case note that the entropy of an intrusion $\mathbf{H}(I)$ is independent of our optimization parameters (P_{FA}, P_D), therefore we have:

$$\begin{aligned}
(P_{FA}^*, P_D^*) &= \arg \max_{(P_{FA},P_D)\in ROC} \frac{\mathbf{I}(I;A)}{\mathbf{H}(I)} \\
&= \arg \max_{(P_{FA},P_D)\in ROC} \mathbf{I}(I;A) \\
&= \arg \min_{(P_{FA},P_D)\in ROC} \mathbf{H}(I|A) \\
&= \arg \min_{(P_{FA},P_D)\in ROC} \mathbf{E}[-\log \Pr[I|A]]
\end{aligned}$$

It is now clear that C_{ID} is an instance of the expected cost problem with costs given by $C(i,j) = -\log \Pr[I = i|A = j]$. By finding the costs of C_{ID} we are making the C_{ID} metric more intuitively appealing, since any optimal point that we find for the IDS will have an explanation in terms of cost functions (as opposed to the vague notion of diminishing the uncertainty of the intrusions).

Finally, in order to combine the PPV and the NPV in an average cost metric, recall that we want to maximize both $\Pr[I = 1|A = 1]$ and $\Pr[I = 0|A = 0]$. Our average gain for each operating point of the IDS is therefore

$$\omega_1 \Pr[I = I|A = 1]\Pr[A = 1] + \omega_2 \Pr[I = 0|A = 0]\Pr[A = 0]$$

where ω_1 (ω_2) is a weight representing a preference towards maximizing PPV (NPV). This equation is equivalent to the minimization of

$$-\omega_1 \Pr[I = I|A = 1]\Pr[A = 1] - \omega_2 \Pr[I = 0|A = 0]\Pr[A = 0] \tag{2.7}$$

Comparing equation (2.7) with equation (2.2), we identify the costs as being $C(1,1) = -\omega_1$, $C(0,0) = -\omega_2$ and $C(0,1) = C(1,0) = 0$. Relating the predictive value metrics (PPV and NPV) with the expected cost problem will allow us to examine the effects of the base-rate fallacy on the expected cost of the IDS in future sections.

IDS classification tradeoffs

An alternate approach in evaluating intrusion detection systems is to directly compare the tradeoffs in the operation of the system by a tradeoff curve, such as ROC, or DET curves [18] (a reinterpretation of the ROC curve where the y-axis is $1 - P_D$, as opposed to P_D). As mentioned in [7], another tradeoff to consider is between the PPV and the NPV values. However, we do not know of any tradeoff curves that combine these two values to aid the operator in choosing a given operating point.

We point out in section B that a tradeoff between P_{FA} and P_D (as in the ROC curves) as well as a tradeoff between PPV and NPV can be misleading for cases where p is very small, since very small changes in the P_{FA} and NPV values for our points of interest will have drastic performance effects on the P_D and the PPV values. Therefore, in the next section we introduce the B-ROC as a new tradeoff curve between P_D and PPV.

The class imbalance problem

A way to relate our approach with traditional techniques used in machine learning is to identify the base-rate fallacy as just another instance of the class imbalance problem. The term *class imbalance* refers to the case when in a classification task, there are many more instances of some classes than others. The *problem* is that under this setting, classifiers in general perform poorly because they tend to concentrate on the large classes and disregard the ones with few examples.

Given that this problem is prevalent in a wide range of practical classification problems, there has been recent interest in trying to design and evaluate classifiers faced with imbalanced data sets [19, 20, 21].

A number of approaches on how to address these issues have been proposed in the literature. Ideas such as data sampling methods, one-class learning (i.e. recognition-based learning), and feature selection algorithms, appear to be the most active research directions for learning classifiers. On the other hand the issue of how to evaluate *binary* classifiers in the case of class imbalances appears to be dominated by the use of ROC curves [22, 23] (and to a lesser extent, by error curves [24]).

V Graphical Analysis

We now introduce a graphical framework that allows the comparison of different metrics in the analysis and evaluation of IDSs. This graphical framework can be used to adaptively change the parameters of the IDS based on its actual performance during operation. The framework also allows for the comparison of different IDSs under different operating environments.

Throughout this section we use one of the ROC curves analyzed in [8] and in [9]. Mainly the ROC curve describing the performance of the COLUMBIA team intrusion detector for the 1998 DARPA intrusion detection evaluation [25]. Unless otherwise stated, we assume for our analysis the base-rate present in the DARPA evaluation which was $p = 6.52 \times 10^{-5}$.

A. *Visualizing the Expected Cost: The Minimization Approach*

The biggest drawback of the expected cost approach is that the assumptions and information about the likelihood of attacks and costs might not be known a priori. Moreover, these parameters can change dynamically during the system operation. It is thus desirable to be able to tune the uncertain IDS parameters based on feedback from its actual system performance in order to minimize $\mathbf{E}[C(I,A)]$.

We select the use of ROC curves as the basic 2-D graph because they illustrate the behavior of a classifier without regard to the uncertain parameters, such as the base-rate p and the operational costs $C(i,j)$. Thus the ROC curve decouples the classification performance from these factors [26]. ROC curves are also general enough such that they can be used to study anomaly detection schemes and misuse detection schemes (a misuse detection scheme has only one point in the ROC space).

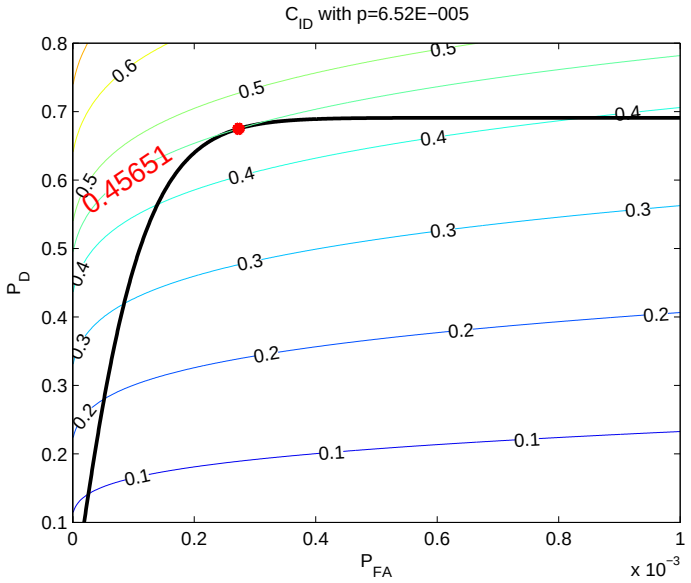


Figure 2.1: Isoline projections of C_{ID} onto the ROC curve. The optimal C_{ID} value is $C_{ID} = 0.4565$. The associated costs are $C(0,0) = 3 \times 10^{-5}$, $C(0,1) = 0.2156$, $C(1,0) = 15.5255$ and $C(1,1) = 2.8487$. The optimal operating point is $P_{FA} = 2.76 \times 10^{-4}$ and $P_D = 0.6749$.

In the graphical framework, the relation of these uncertain factors with the ROC curve of an IDS will be reflected in the *isolines* of each metric, where isolines refer to lines that connect pairs of false alarm and detection rates such that any point on the line has equal expected cost. The evaluation of an IDS is therefore reduced to finding the point of the ROC curve that intercepts the optimal isoline of the metric (for signature detectors the evaluation corresponds to finding the isoline that intercepts their single point in the ROC space and the point (0,0) or (1,1)). In Figure 2.1 we can see as an example the isolines of C_{ID} intercepting the ROC curve of the 1998 DARPA intrusion detection evaluation.

One limitation of the C_{ID} metric is that it specifies the costs $C(i,j)$ a priori. However, in practice these costs are rarely known in advance and moreover the costs can change and be dynamically adapted based on the performance of the IDS. Furthermore the nonlinearity of C_{ID} makes it difficult to analyze the effect different p values will have on C_{ID} in a single 2-D graph. To make the graphical analysis of the cost metrics as intuitive as possible, we will assume from

now on (as in [8]) that the costs are tunable parameters and yet once a selection of their values is made, they are constant values. This new assumption will let us at the same time see the effect of different values of p in the expected cost metric.

Under the assumption of constant costs, we can see that the isolines for the expected cost $\mathbf{E}[C(I,A)]$ are in fact straight lines whose slope depends on the ratio between the costs and the likelihood ratio of an attack. Formally, if we want the pair of points (P_{FA1}, P_{D1}) and (P_{FA2}, P_{D2}) to have the same expected cost, they must be related by the following equation [27, 28, 26]:

$$m_{C,p} \equiv \frac{P_{D2} - P_{D1}}{P_{FA1} - P_{FA2}} = \frac{1-p}{p} \frac{C(0,1) - C(0,0)}{C(1,0) - C(1,1)} = \frac{1-p}{p} \frac{1}{C} \tag{2.8}$$

where in the last equality we have implicitly defined C to be the ratio between the costs, and $m_{C,p}$ to be the slope of the isoline. The set of isolines of $\mathbf{E}[C(I,A)]$ can be represented by

$$ISO_E = \{m_{C,p} \times P_{FA} + b : b \in [0,1]\} \tag{2.9}$$

For fixed C and p, it is easy to prove that the optimal operating point of the ROC is the point where the ROC intercepts the isoline in ISO_E with the largest b (note however that there are ROC curves that can have more than one optimal point.) The optimal operating point in the ROC is therefore determined only by the slope of the isolines, which in turn is determined by p and C. Therefore we can readily check how changes in the costs and in the likelihood of an attack will impact the optimal operating point.

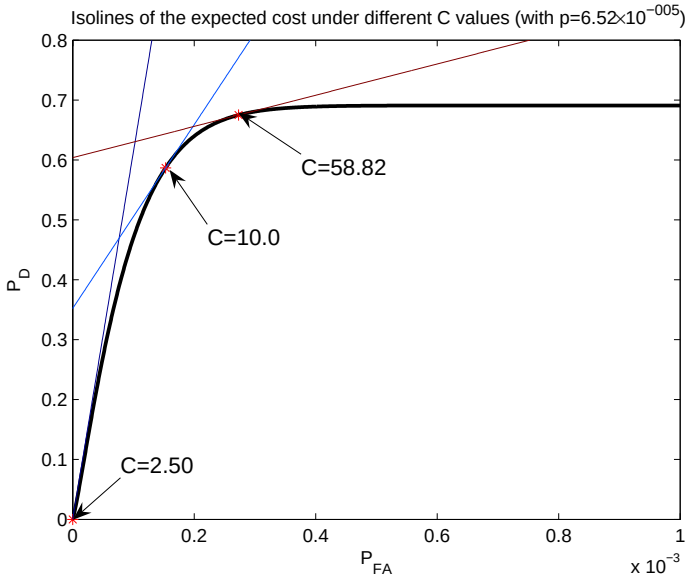


Figure 2.2: As the cost ratio C increases, the slope of the optimal isoline decreases

Effect of the Costs

In Figure 2.2, consider the operating point corresponding to $C = 58.82$, and assume that after some time, the operators of the IDS realize that the number of false alarms exceeds their response capabilities. In order to reduce the number of false alarms they can increase the cost of a false alarm $C(0,1)$ and obtain a second operating point at $C = 10$. If however the situation

persists (i.e. the number of false alarms is still much more than what operators can efficiently respond to) and therefore they keep increasing the cost of a false alarm, there will be a *critical slope* m^c such that the intersection of the ROC and the isoline with slope m^c will be at the point $(P_{FA}, P_D) = (0,0)$. The interpretation of this result is that we should not use the IDS being evaluated since its performance is not good enough for the environment it has been deployed in. In order to solve this problem we need to either change the environment (e.g. hire more IDS operators) or change the IDS (e.g. shop for a more expensive IDS).

The Base-Rate Fallacy Implications on the Costs of an IDS

A similar scenario occurs when the likelihood of an attack changes. In Figure 2.3 we can see how as p decreases, the optimal operating point of the IDS tends again to $(P_{FA}, P_D) = (0,0)$ (again the evaluator must decide not to use the IDS for its current operating environment). Therefore, for small base-rates the operation of an IDS will be cost efficient only if we have an appropriate large C^* such that $m_{C^*,p^*} \leq m^c$. A large C^* can be explained if cost of a false alarm much smaller than the cost of a missed detection: $C(1,0) \gg C(0,1)$ (e.g. the case of a government network that cannot afford undetected intrusions and has enough resources to sort through the false alarms).

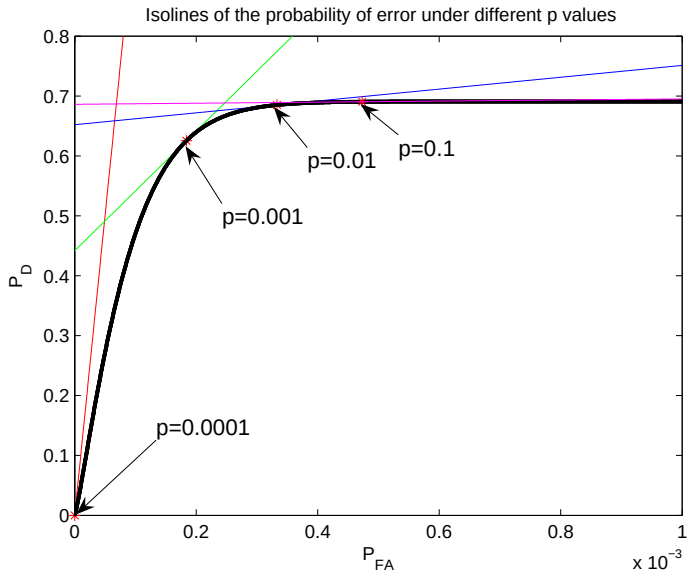


Figure 2.3: As the base-rate p decreases, the slope of the optimal isoline increases

Generalizations

This graphical method of cost analysis can also be applied to other metrics in order to get some insight into the expected cost of the IDS. For example in [10], the authors define an IDS with input space $\mathcal{X}$ to be $\sigma - sensitive$ if there exists an efficient algorithm with the same input space $\mathcal{E} : \mathcal{X} \rightarrow \{\neg A, A\}$, such that $P_D^{\mathcal{E}} - P_{FA}^{\mathcal{E}} \geq \sigma$. This metric can be used to find the optimal point of an ROC because it has a very intuitive explanation: as long as the rate of detected intrusions increases faster than the rate of false alarms, we keep moving the operating point

of the IDS towards the right in the ROC. The optimal sensitivity problem for an IDS with a receiver operating characteristic *ROC* is thus:

$$\max_{(P_{FA},P_D)\in ROC} P_D - P_{FA} \tag{2.10}$$

It is easy to show that this optimal sensitivity point is the same optimal point obtained with the isolines method for $m_{C,p} = 1$ (i.e. $C = (1-p)/p$).

B. The Bayesian Receiver Operating Characteristic: The Tradeoff Approach

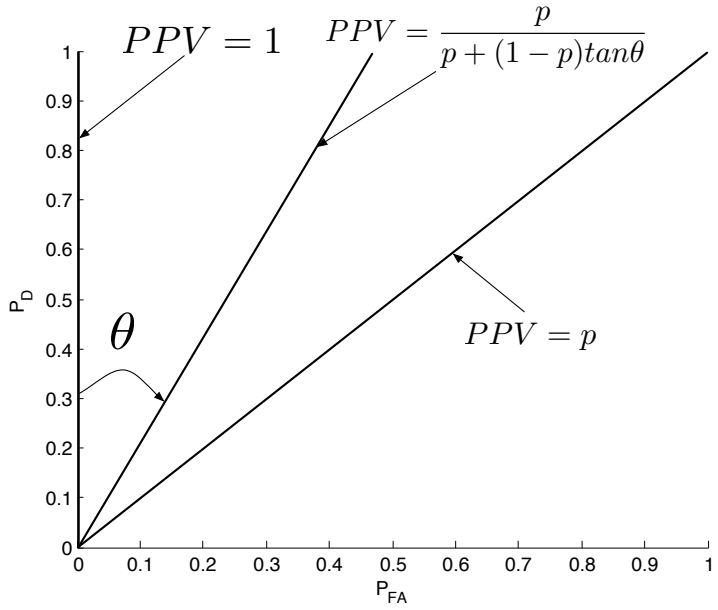


Figure 2.4: The PPV isolines in the ROC space are straight lines that depend only on θ. The PPV values of interest range from 1 to p

Although the graphical analysis introduced so far can be applied to analyze the cost efficiency of several metrics, the intuition for the tradeoff between the PPV and the NPV is still not clear. Therefore we now extend the graphical approach by introducing a new pair of isolines, those of the PPV and the NPV metrics.

Lemma 1: Two sets of points (P_{FA1}, P_{D1}) and (P_{FA2}, P_{D2}) have the same PPV value if and only if

$$\frac{P_{FA2}}{P_{D2}} = \frac{P_{FA1}}{P_{D1}} = \tan\theta \tag{2.11}$$

where θ is the angle between the line $P_{FA} = 0$ and the isoline. Moreover the PPV value of an isoline at angle θ is

$$PPV_{\theta,p} = \frac{p}{p + (1-p)\tan\theta} \tag{2.12}$$

Similarly, two set of points (P_{FA1}, P_{D1}) and (P_{FA2}, P_{D2}) have the same NPV value if and only if

$$\frac{1-P_{D1}}{1-P_{FA1}} = \frac{1-P_{D2}}{1-P_{FA2}} = \tan\phi \tag{2.13}$$

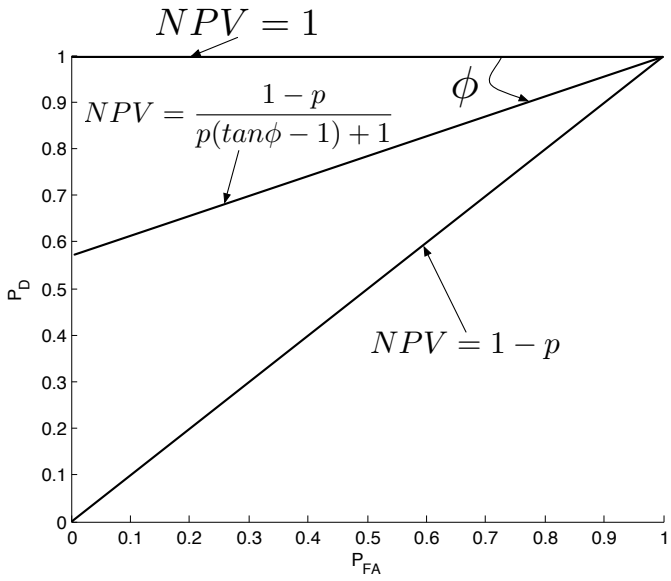


Figure 2.5: The NPV isolines in the ROC space are straight lines that depend only on ϕ. The NPV values of interest range from 1 to $1-p$

where ϕ is the angle between the line $P_D = 1$ and the isoline. Moreover the NPV value of an isoline at angle ϕ is

$$NPV_{\phi,p} = \frac{1-p}{p(\tan\phi - 1) + 1} \tag{2.14}$$

Figures 2.4 and 2.5 show the graphical interpretation of Lemma 1. It is important to note the range of the PPV and NPV values as a function of their angles. In particular notice that as θ goes from 0° to 45° (the range of interest), the value of PPV changes from 1 to p. We can also see from Figure 2.5 that as ϕ ranges from 0° to 45°, the NPV value changes from one to $1-p$. If p is very small, then $NPV \approx 1$.

Figure 2.6 shows the application of Fact 1 to a typical ROC curve of an IDS. In this figure we can see the tradeoff of four variables of interest: P_{FA}, P_D, PPV, and NPV. Notice that if we choose the optimal operating point based on P_{FA} and P_D, as in the typical ROC analysis, we might obtain misleading results because we do not know how to interpret intuitively very low false alarm rates, e.g. is $P_{FA} = 10^{-3}$ much better than $P_{FA} = 5 \times 10^{-3}$? The same reasoning applies to the study of PPV vs. NPV as we cannot interpret precisely small variations in NPV values, e.g. is $NPV = 0.9998$ much better than $NPV = 0.99975$? Therefore we conclude that the most relevant metrics to use for a tradeoff in the performance of a classifier are P_D and PPV, since they have an easily understandable range of interest.

However, even when you select as tradeoff parameters the PPV and P_D values, the isoline analysis shown in Figure 2.6 has still one deficiency, and it is the fact that there is no efficient way to account for the uncertainty of p. In order to solve this problem we introduce the B-ROC as a graph that shows how the two variables of interest: P_D and PPV are related under different severity of class imbalances. In order to follow the intuition of the ROC curves, instead of using PPV for the x-axis we prefer to use 1-PPV. We use this quantity because it can be interpreted as the *Bayesian false alarm rate*: $B_{FA} \equiv \Pr[C = 0|A = 1]$. For example, for IDSs B_{FA} can be a

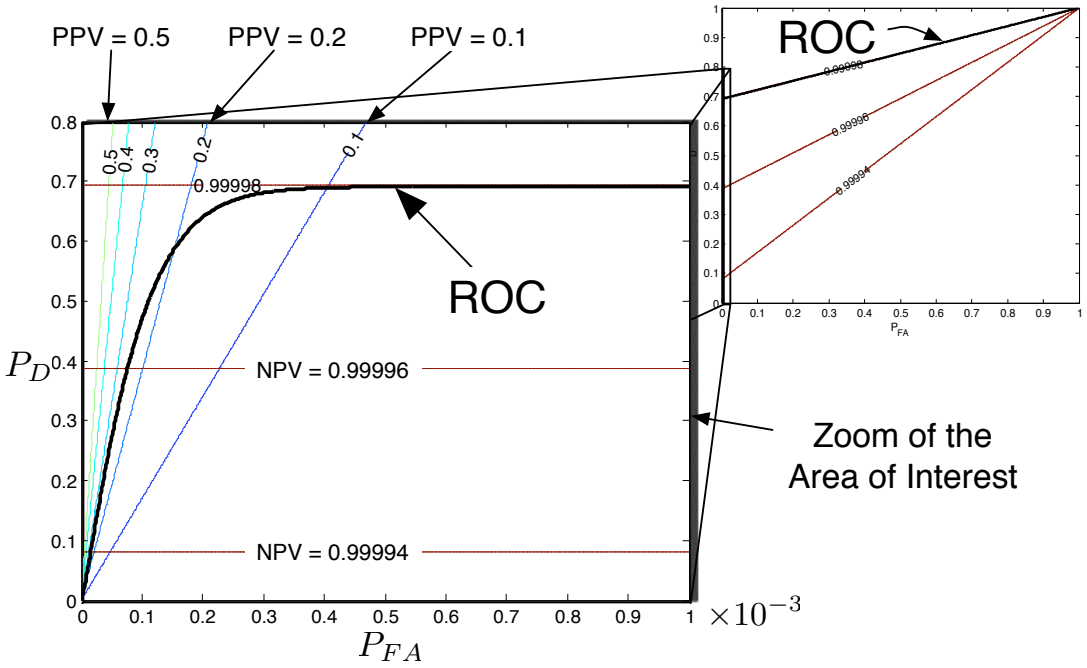


Figure 2.6: PPV and NPV isolines for the ROC of an IDS with $p = 6.52 \times 10^{-5}$

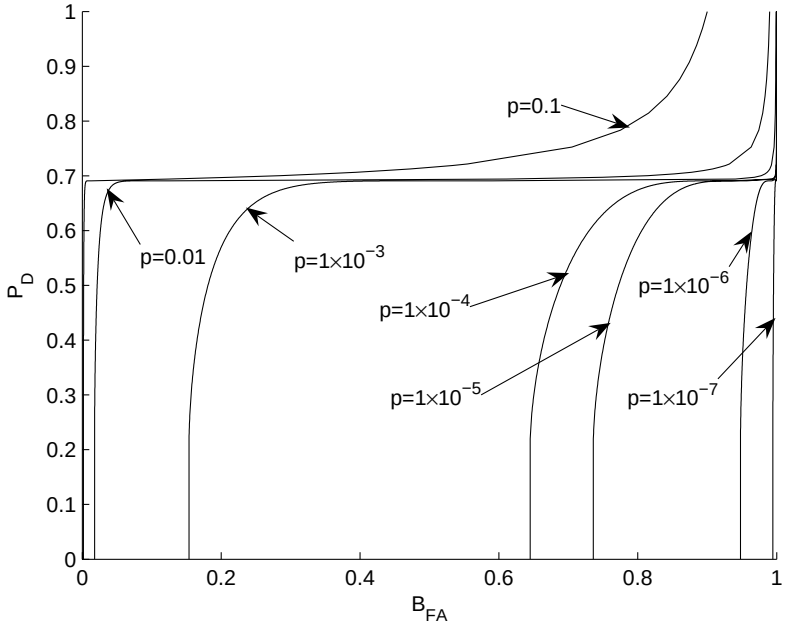


Figure 2.7: B-ROC for the ROC of Figure 2.6.

measure of how likely it is, that the operators of the detection system will loose their time each time they respond to an alarm. Figure 2.7 shows the B-ROC for the ROC presented in Figure 2.6. Notice also how the values of interest for the x-axis have changed from $[0, 10^{-3}]$ to $[0, 1]$.

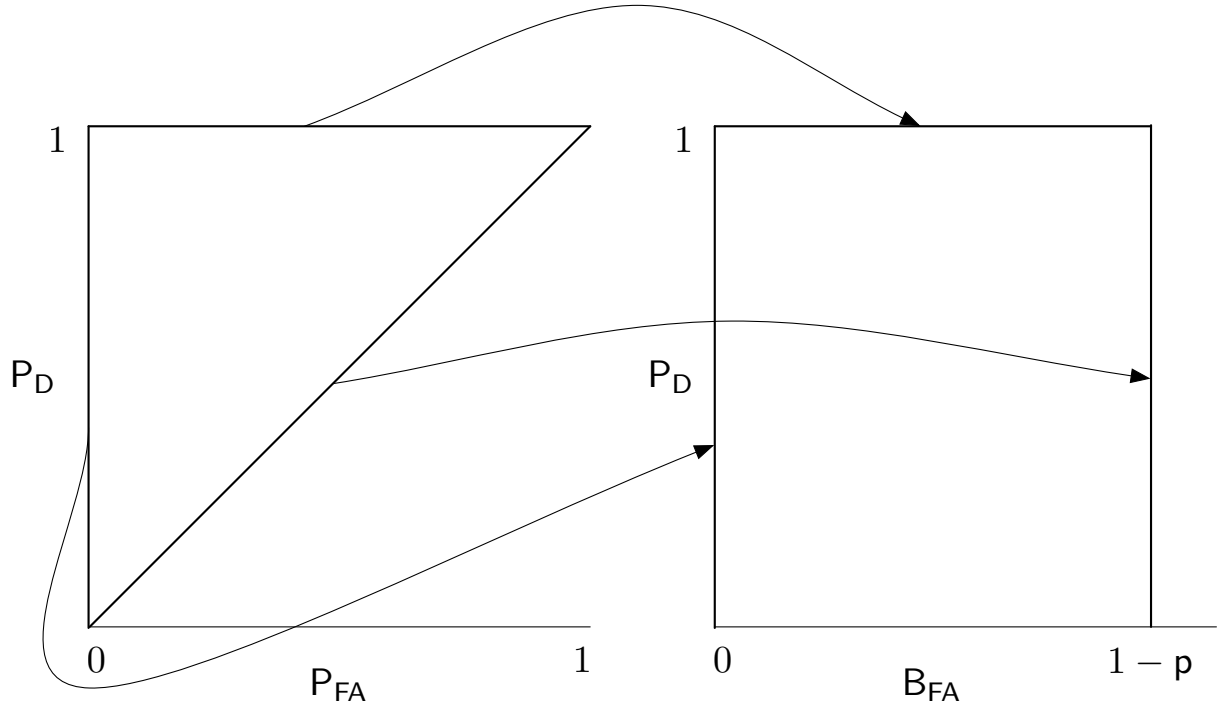


Figure 2.8: Mapping of ROC to B-ROC

In order to be able to interpret the B-ROC curves, Figure 2.8 shows how the ROC points map to points in the B-ROC. The vertical line defined by $0 < P_D \leq 1$ and $P_{FA} = 0$ in the ROC maps exactly to the same vertical line $0 < P_D \leq 1$ and $B_{FA} = 0$ in the B-ROC. Similarly, the top horizontal line $0 \leq P_{FA} \leq 1$ and $P_D = 1$ maps to the line $0 \leq B_{FA} \leq 1 - p$ and $P_D = 1$. A classifier that performs random guessing is represented in the ROC as the diagonal line $P_D = P_{FA}$, and this random guessing classifier maps to the vertical line defined by $B_{FA} = 1 - p$ and $P_D > 0$ in the B-ROC. Finally, to understand where the point $(0,0)$ in the ROC maps into the B-ROC, let α and $f(\alpha)$ denote P_{FA} and the corresponding P_D in the ROC curve. Then, as the false alarm rate α tends to zero (from the right), the Bayesian false alarm rate tends to a value that depends on p and the slope of the ROC close to the point $(0,0)$. More specifically, if we let $f'(0^+) = \lim_{\alpha \to 0^+} f'(\alpha)$, then:

$$\lim_{\alpha \to 0^+} B_{FA} = \lim_{\alpha \to 0^+} \frac{\alpha(1-p)}{p f(\alpha) + \alpha(1-p)} = \frac{1-p}{p(f'(0^+) - 1) + 1}$$

It is also important to recall that a necessary condition for a classifier to be optimal, is that its ROC curve should be concave. In fact, given any non-concave ROC, by following Neyman-Pearson theory, you can always get a concave ROC curve by randomizing decisions between optimal points [27]. This idea has been recently popularized in the machine learning community by the notion of the ROC convex hull [26].

The importance of this observation is that in order to guarantee that the B-ROC is a well defined continuous and non-decreasing function, we map only concave ROC curves to B-ROCs.

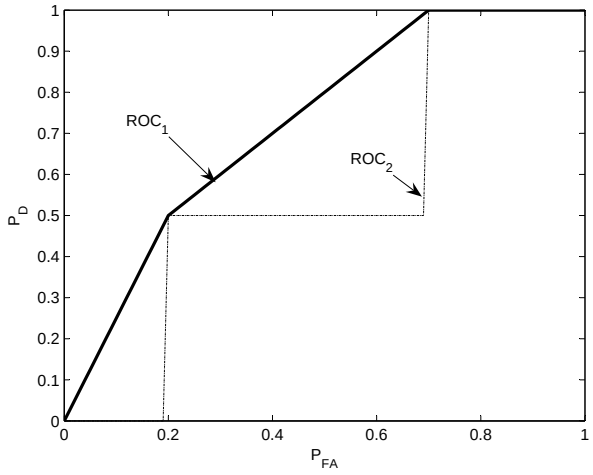


Figure 2.9: An empirical ROC (ROC_2) and its convex hull (ROC_1)

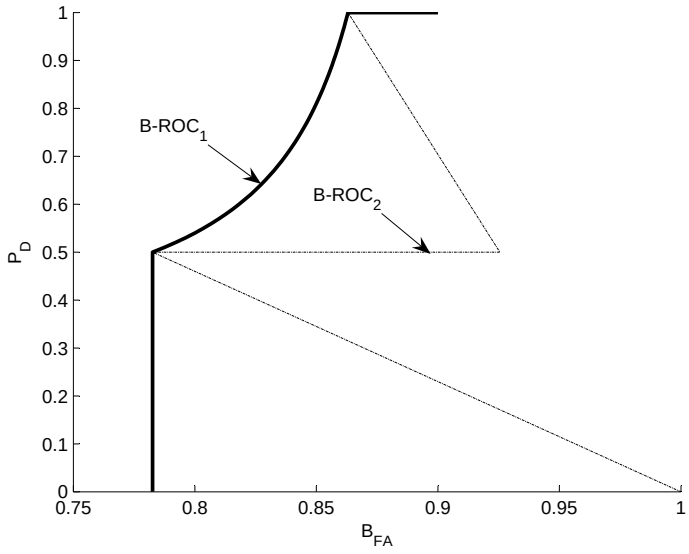


Figure 2.10: The B-ROC of the concave ROC is easier to interpret

In Figures 2.9 and 2.10 we show the only example in this document of the type of B-ROC curve that you can get when you do not consider a concave ROC.

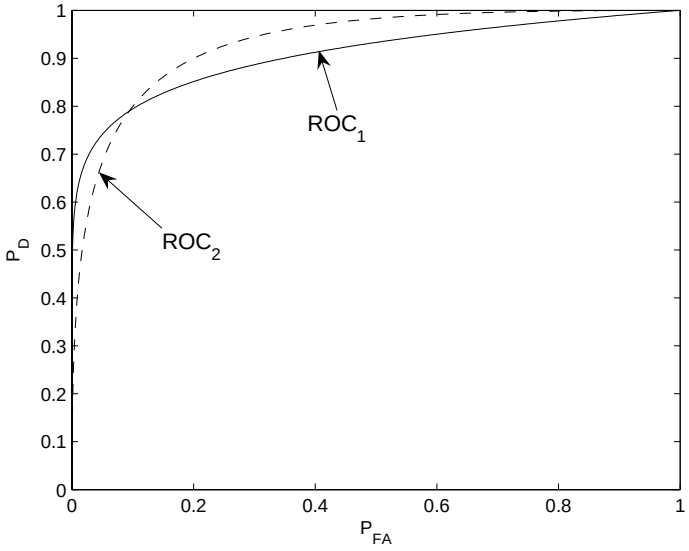


Figure 2.11: Comparison of two classifiers

We also point out the fact that the B-ROC curves can be very useful for the comparison of classifiers. A typical comparison problem by using ROCs is shown in Figure 2.11. Several ideas have been proposed in order to solve this comparison problem. For example by using decision theory as in [29, 26] we can find the optimal classifier between the two by assuming a given prior p and given misclassification costs. However, a big problem with this approach is that the misclassification costs are sometimes uncertain and difficult to estimate a priori. With a B-ROC on the other hand, you can get a better comparison of two classifiers without the assumption of any misclassification costs, as can be seen in Figure 2.12.

VI Conclusions

We believe that the B-ROC provides a better way to evaluate and compare classifiers in the case of class imbalances or uncertain values of p. First, for selecting operating points in heavily imbalanced environments, B-ROCs use tradeoff parameters that are easier to understand than the variables considered in ROC curves (they provide better intuition for the performance of the classifier). Second, since the exact class distribution p might not be known a priori, or accurately enough, the B-ROC allows the plot of different curves for the range of interest of p. Finally, when comparing two classifiers, there are cases in which by using the B-ROC, we do not need cost values in order to decide which classifier would be better for given values of p. Note also that B-ROCs consider parameters that are directly related to exact quantities that the operator of a classifier can measure. In contrast, the exact interpretation of the expected cost of a classifier is more difficult to relate to the real performance of the classifier (the costs depend in many other unknown factors).

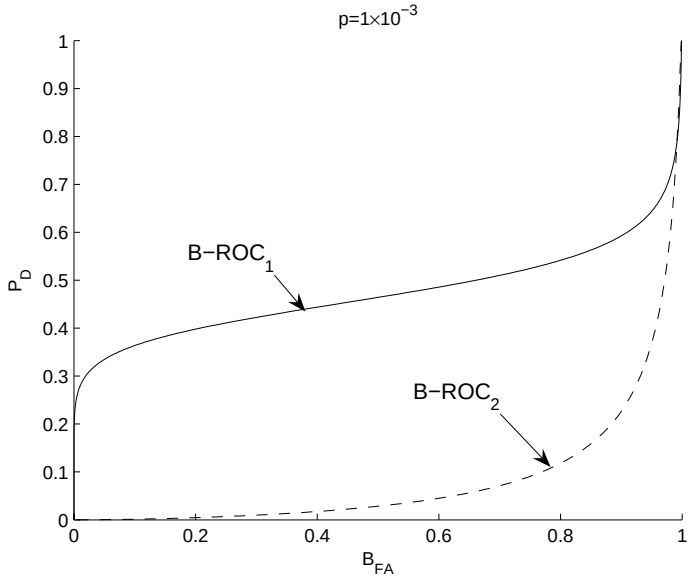


Figure 2.12: B-ROCs comparison for the *p* of interest

Chapter 3
Secure Decision Making: Defining the Evaluation Metrics and the Adversary

They did not realize that because of the quasi-reciprocal and circular nature of all Improbability calculations, anything that was Infinitely Improbable was actually very likely to happen almost immediately.
-Life the Universe and Everything. Douglas Adams

I Overview

As we have seen in the previous chapter, the traditional evaluation of IDSs assume that the intruder will behave similarly before and after the deployment and configuration of the IDS (i.e. during the evaluation it is assumed that the intruder will be non-adaptive). In practice however this assumption does not hold, since once an IDS is deployed, intruders will adapt and try to evade detection or launch attacks against the IDS.

This lack of a proper adversarial model is a big problem for the evaluation of any decision making algorithm used in security applications, since without the definition of an adversary, we cannot reason about and much less measure the security of the algorithm.

We therefore layout in this chapter the overall design and evaluation goals that will be used throughout the rest of this dissertation.

The basic idea is to introduce a formal framework for reasoning about the performance and security of a decision making algorithm. In particular we do not want to find or design the best performing decision making algorithm on average, but the algorithm that performs best against the worst type of attacks.

The chapter layout is the following. In section II we introduce a general set of guidelines for the design and evaluation of decision making algorithms. In section III we introduce a black box adversary model. This model is very simple yet very useful and robust for cases where having an exact adversarial model is intractable. Finally in section IV we introduce a detailed model of an adversary which we will use in chapters 4 and 5.

II A Set of Design and Evaluation Guidelines

In this chapter we focus on designing a practical methodology for dealing with attackers. In particular we propose the use of a framework where each of these components is clearly defined:

Desired Properties Intuitive definition of the goal of the system.

Feasible Design Space The design space $\mathcal{S}$ for the classification algorithm.

Information Available to the Adversary Identify which pieces of information can be available to an attacker.

Capabilities of the Adversary Define a feasible class of attackers $\mathcal{F}$ based on the assumed capabilities.

Evaluation Metric The evaluation metric should be a reasonable measure how well the designed system meets our desired properties. We call a system *secure* if its metric outcome is satisfied for any feasible attacker.

Objective of the Adversary An attacker can use its capabilities and the information available in order to perform two main classes of attacks:

- *Evaluation attack*. The goal of the attacker is opposite to the goal defined by the evaluation metric. For example, if the goal of the classifier is to minimize $\mathbb{E}[L(C,A)]$, then the goal of the attacker is to maximize $\mathbb{E}[L(C,A)]$.
- *Base system attack*. The goal of an attacker is not the opposite goal of the classifier. For example, even if the goal of the classifier is to minimize $\mathbb{E}[L(C,A)]$, the goal of the attacker is still to minimize the probability of being detected.

Model Assumptions Identify clearly the assumptions made during the design and evaluation of the classifier. It is important to realize that when we borrow tools from other fields, they come with a set of assumptions that might not hold in an adversarial setting, because the first thing that an attacker will do is violate the set of assumptions that the classifier is relying on for proper operation. After all, the UCI machine learning repository never launched an attack against your classifier. Therefore one of the most important ways to deal with an adversarial environment is to limit the number of assumptions made, and to evaluate the resiliency of the remaining assumptions to attacks.

Before we use these guidelines for the evaluation of classifiers, we describe a very simple example of the use of these guidelines and their relationship to cryptography. Cryptography is one of the best examples in which a precise framework has been developed in order to define properly what a secure system means, and how to model an adversary. We believe therefore that this example will help us in identifying the generality and use of the guidelines as a step towards achieving sound designs[1].

A. Example: Secret key Encryption

In secret key cryptography, Alice and Bob share a single key *sk*. Given a message *m* (called *plaintext*) Alice uses an encryption algorithm to produce unintelligible data *C* (called *ciphertext*): $C \leftarrow \mathcal{E}_{sk}(m)$. After receiving *C*, Bob then uses *sk* and a decryption algorithm to recover the secret message $m = \mathcal{D}_{sk}(C)$.

Desired Properties $\mathcal{E}$ and $\mathcal{D}$ should enable Alice and Bob to communicate secretly, that is, a feasible adversary should not get any information about *m* given *C* except with very small probability.

Feasible Design Space $\mathcal{E}$ and $\mathcal{D}$ have to be efficient probabilistic algorithms. They also need to satisfy correctness: for any *sk* and *m*, $\mathcal{D}_{sk}(\mathcal{E}_{sk}(m)) = m$.

[1]We avoid the precise formal treatment of cryptography because our main objective here is to present the intuition behind the principles rather than the specific technical details.

Information Available to the Adversary It is assumed that an adversary knows the encryption and decryption algorithms. The only information not available to the adversary is the secret key *sk* shared between Alice and Bob.

Capabilities of the Adversary The class of feasible adversaries $\mathcal{F}$ is the set of algorithms running in a reasonable amount of time.

Evaluation Metric For any messages m_0 and m_1, given a ciphertext C which is known to be an encryption of either m_0 or m_1, no adversary $\mathcal{A} \in \mathcal{F}$ can guess correctly which message was encrypted with probability significantly better than $1/2$.

Goal of the Adversary Perform an evaluation attack. That is, design an algorithm $\mathcal{A} \in \mathcal{F}$ that can guess with probability significantly better than $1/2$ which message corresponds to the given ciphertext.

Model Assumptions The security of an encryption scheme usually relies in a set of cryptographic primitives, such as one way functions.

Another interesting aspect of cryptography is the different notions of security when the adversary is modified. In the previous example it is sometimes reasonable to assume that the attacker will obtain valid plaintext and ciphertext pairs: $\{(m_0, C_0), (m_1, C_1), \ldots, (m_k, C_k)\}$. This new setting is modeled by giving the adversary more capabilities: the feasible set $\mathcal{F}$ will now consist of all efficient algorithms that have access to the ciphertexts of chosen plaintexts. An encryption algorithm is therefore secure against *chosen-ciphertext* attacks if even with this new capability, the adversary still cannot break the encryption scheme.

III A Black Box Adversary Model

In this section we introduce one of the simplest adversary models against classification algorithms. We refer to it as a *black box* adversary model since we do not care at the moment how the adversary creates the observations X. This model is particularly suited to cases where trying to model the creation of the inputs to the classifier is intractable.

Recall first that for our evaluation analysis in the previous chapter we were assuming three quantities that can be, up to a certain extent, controlled by the intruder. They are the base-rate p, the false alarm rate P_{FA} and the detection rate P_D. The base-rate can be modified by controlling the frequency of attacks. The perceived false alarm rate can be increased if the intruder finds a flaw in any of the signatures of an IDS that allows him to send maliciously crafted packets that trigger alarms at the IDS but that look benign to the IDS operator. Finally, the detection rate can be modified by the intruder with the creation of new attacks whose signatures do not match those of the IDS, or simply by evading the detection scheme, for example by the creation of a mimicry attack [30, 31].

In an effort towards understanding the advantage an intruder has by controlling these parameters, and to provide a robust evaluation framework, we now present a formal framework to reason about the robustness of an IDS evaluation method. Our work in this section is in some sense similar to the theoretical framework presented in [10], which was inspired by cryptographic models. However, we see two main differences in our work. First, we introduce the role of an adversary against the IDS, and thereby introduce a measure of robustness for the metrics. In the second place, our work is more practical and is applicable to more realistic

evaluation metrics. Furthermore we also provide examples of practical scenarios where our methods can be applied.

In order to be precise in our presentation, we need to extend the definitions introduced in section III. For our modeling purposes we decompose the $I\mathcal{DS}$ algorithm into two parts: a *detector* $\mathcal{D}$ and a *decision maker* $\mathcal{DM}$. For the case of an anomaly detection scheme, $\mathcal{D}(\mathbf{x}[j])$ outputs the anomaly score $y[j]$ on input $\mathbf{x}[j]$ and $\mathcal{DM}$ represents the threshold that determines whether to consider the anomaly score as an intrusion or not, i.e. $\mathcal{DM}(y[j])$ outputs an alarm or it does not. For a misuse detection scheme, $\mathcal{DM}$ has to decide to use the signature to report alarms or decide that the performance of the signature is not good enough to justify its use and therefore will ignore all alarms (e.g. it is not cost-efficient to purchase the misuse scheme being evaluated).

Definition 1 An $I\mathcal{DS}$ algorithm is the composition of algorithms $\mathcal{D}$ (an algorithm from where we can obtain an ROC curve) and $\mathcal{DM}$ (an algorithm responsible for selecting an operating point). During operation, an $I\mathcal{DS}$ receives a continuous data stream of event features $\mathbf{x}[1], \mathbf{x}[2], \ldots$ and classifies each input $\mathbf{x}[j]$ by raising an alarm or not. Formally:[2]

$$
\begin{array}{l}
\underline{I\mathcal{DS}}(\mathbf{x}) \\
\qquad y = \mathcal{D}(\mathbf{x}) \\
\qquad A \leftarrow \mathcal{DM}(y) \\
\qquad \text{Output } A \text{ (where } A \in \{0,1\})
\end{array}
$$

$\diamondsuit$

We now study the performance of an IDS under an adversarial setting. We remark that our intruder model does not represent a single physical attacker against the IDS. Instead our model represents a collection of attackers whose average behavior will be studied under the worst possible circumstances for the IDS.

The first thing we consider, is the amount of information the intruder has. A basic assumption to make in an adversarial setting is to consider that the intruder knows everything that we know about the environment and can make inferences about the situation the same way as we can. Under this assumption we assume that the base-rate $\hat{p}$ estimated by the IDS, its estimated operating condition $(\hat{P}_{FA}, \hat{P}_D)$ selected during the evaluation, the original *ROC* curve (obtained from $\mathcal{D}$) and the cost function $C(I,A)$ are *public values* (i.e. they are known to the intruder).

We model the capability of an adaptive intruder by defining some confidence bounds. We assume an intruder can deviate $\hat{p} - \delta_l$, $\hat{p} + \delta_u$ from the expected $\hat{p}$ value. Also, based on our confidence in the detector algorithm and how hard we expect it to be for an intruder to evade the detector, or to create non-relevant false positives (this also models how the normal behavior of the system being monitored can produce new -previously unseen- false alarms), we define α and β as bounds to the amount of variation we can expect during the IDS operation from the false alarms and the detection rate (respectively) we expected, i.e. the amount of variation from $(\hat{P}_{FA}, \hat{P}_D)$ (although in practice estimating these bounds is not an easy task, testing approaches like the one described in [32] can help in their determination).

The intruder also has access to an oracle **Feature**$(\cdot,\cdot)$ that simulates an event to input into the IDS. **Feature**$(0,\zeta)$ outputs a feature vector modeling the normal behavior of the system that will raise an alarm with probability ζ (or a crafted malicious feature to only raise alarms in the

[2]The usual arrow notation: $a \leftarrow \mathcal{DM}(y)$ implies that $\mathcal{DM}$ can be a probabilistic algorithm.

case **Feature**$(0,1)$). And **Feature**$(1,\zeta)$ outputs the feature vector of an intrusion that will raise an alarm with probability ζ.

Definition 2 A $(\delta,\alpha,\beta)-$*intruder* is an algorithm $\mathfrak{I}$ that can select its frequency of intrusions p_1 from the interval $\delta=[\hat{p}-\delta_l,\hat{p}+\delta_u]$. If it decides to attempt an intrusion, then with probability $p_2\in[0,\beta]$, it creates an attack feature $\mathbf{x}$ that will go undetected by the IDS (otherwise this intrusion is detected with probability $\hat{P}_D$). If it decides not to attempt an intrusion, with probability $p_3\in[0,\alpha]$ it creates a feature $\mathbf{x}$ that will raise a false alarm in the IDS

$\underline{\mathfrak{I}}(\delta,\alpha,\beta)$
Select $p_1\in[\hat{p}-\delta_l,\hat{p}+\delta_u]$
Select $p_2\in[0,\alpha]$
Select $p_3\in[0,\beta]$
$I\leftarrow$ Bernoulli(p_1)
If $I=1$
$B\leftarrow$ Bernoulli(p_3)
$\mathbf{x}\leftarrow$ **Feature**$(1,(min\{(1-B),\hat{P}_D\}))$
Else
$B\leftarrow$ Bernoulli(p_2)
$\mathbf{x}\leftarrow$ **Feature**$(0,max\{B,\hat{P}_{FA}\})$
Output (I,**x**)

where Bernoulli(ζ) outputs a Bernoulli random variable with probability of success ζ.

Furthermore, if $\delta_l=p$ and $\delta_u=1-p$ we say that $\mathfrak{I}$ has the ability to make a *chosen-intrusion rate attack.*

$\diamondsuit$

We now formalize what it means for an evaluation scheme to be robust. We stress the fact that we are not analyzing the security of an IDS, but rather the security of the *evaluation* of an IDS, i.e. how confident we are that the IDS will behave during operation similarly to what we assumed in the evaluation.

A. Robust Expected Cost Evaluation

We start with the general decision theoretic framework of evaluating the expected cost (per input) $\mathbb{E}[C(I,A)]$ for an IDS.

Definition 3 An evaluation method that claims the expected cost of an $I\mathcal{DS}$ is at most r is **robust** against a $(\delta,\alpha,\beta)-$*intruder* if the expected cost of $I\mathcal{DS}$ during the attack ($\mathbb{E}^{\delta,\alpha,\beta}[C(I,A)]$) is no larger than r, i.e.

$$\mathbb{E}^{\delta,\alpha,\beta}[C[I,A)]=\sum_{i,a}C(i,a)\times$$
$$\Pr[\ (I,\mathbf{x})\leftarrow\mathfrak{I}(\delta,\alpha,\beta);A\leftarrow I\mathcal{DS}(\mathbf{x}):\ I=i,A=a\ ]\leq r$$

$\diamondsuit$

Now recall that the traditional evaluation framework finds an evaluation value r^* by using equation (2.6). So by finding r^* we are basically finding the best performance of an IDS and claiming the IDS is better than others if r^* is smaller than the evaluation of the other IDSs. In this section we claim that an IDS is better than others if its expected value under the worst

performance is smaller than the expected value under the worst performance of other IDSs. In short

Traditional Evaluation Given a set of IDSs $\{IDS_1, IDS_2, \ldots, IDS_n\}$ find the best expected cost for each:

$$r_i^* = \min_{(P_{FA}^{\alpha}, P_D^{\beta}) \in ROC_i} \mathbb{E}[C(I,A)] \tag{3.1}$$

Declare that the best IDS is the one with smallest expected cost r_i^*.

Robust Evaluation Given a set of IDSs $\{IDS_1, IDS_2, \ldots, IDS_n\}$ find the best expected cost for each IDS_i when being under the attack of a $(\delta, \alpha_i, \beta_i) - intruder$[3]. Therefore we find the best IDS as follows:

$$r_i^{robust} = \min_{(P_{FA}^{\alpha_i}, P_D^{\beta_i}) \in ROC_i^{\alpha_i,\beta_i}} \max_{I(\delta,\alpha_i,\beta_i)} \mathbb{E}^{\delta,\alpha_i,\beta_i}[C(I,A)] \tag{3.2}$$

Several important questions can be raised by the above framework. In particular we are interested in finding the least upper bound r such that we can claim the evaluation of IDS to be *robust*. Another important question is how can we design an evaluation of IDS satisfying this least upper bound? Solutions to these questions are partially based on game theory.

Lemma 2: Given an initial estimate of the base-rate $\hat{p}$, an initial ROC curve obtained from $\mathcal{D}$, and constant costs $C(I,A)$, the least upper bound r such that the expected cost evaluation of IDS is r-robust is given by

$$r = R(0, \hat{P}_{FA}^{\alpha})(1 - \hat{p}^{\delta}) + R(1, \hat{P}_D^{\beta})\hat{p}^{\delta} \tag{3.3}$$

where

$$R(0, \hat{P}_{FA}^{\alpha}) \equiv [C(0,0)(1 - \hat{P}_{FA}^{\alpha}) + C(0,1)\hat{P}_{FA}^{\alpha}] \tag{3.4}$$

is the expected cost of IDS under no intrusion and

$$R(1, \hat{P}_D^{\beta}) \equiv [C(1,0)(1 - \hat{P}_D^{\beta}) + C(1,1)\hat{P}_D^{\beta}] \tag{3.5}$$

is the expected cost of IDS under an intrusion, and $\hat{p}^{\delta}$, $\hat{P}_{FA}^{\alpha}$ and $\hat{P}_D^{\beta}$ are the solution to a zero-sum game between the intruder (the maximizer) and the IDS (the minimizer), whose solution can be found in the following way:

1. *Let (P_{FA}, P_D) denote any points of the initial ROC obtained from $\mathcal{D}$ and let $ROC^{(\alpha,\beta)}$ be the ROC curve defined by the points $(P_{FA}^{\alpha}, P_D^{\beta})$, where $P_D^{\beta} = P_D(1-\beta)$ and $P_{FA}^{\alpha} = \alpha + P_{FA}(1-\alpha)$.*

2. *Using $\hat{p} + \delta_u$ in the isoline method, find the optimal operating point (x_u, y_u) in $ROC^{(\alpha,\beta)}$ and using $\hat{p} - \delta_l$ in the isoline method, find the optimal operating point (x_l, y_l) in $ROC^{(\alpha,\beta)}$.*

[3]Note that different IDSs might have different α and β values. For example if IDS_1 is an anomaly detection scheme then we can expect that the probability that new normal events will generate alarms α_1 is larger than the same probability α_2 for a misuse detection scheme IDS_2.

3. *Find the points* (x^*, y^*) *in* $ROC^{(\alpha,\beta)}$ *that intersect the line*

$$y = \frac{C(1,0) - C(0,0)}{C(1,0) - C(1,1)} + x\frac{C(0,0) - C(0,1)}{C(1,0) - C(1,1)}$$

(under the natural assumptions $C(1,0) > R(0,x^*) > C(0,0)$, $C(0,1) > C(0,0)$ *and* $C(1,0) > C(1,1)$*). If there are no points that intersect this line, then set* $x^* = y^* = 1$.

4. *If* $x^* \in [x_l, x_u]$ *then find the base-rate parameter* p^* *such that the optimal isoline of Equation (2.9) intercepts* $ROC^{(\alpha,\beta)}$ *at* (x^*, y^*) *and set* $\hat{p}^\delta = p^*$, $\hat{P}^\alpha_{FA} = x^*$ *and* $\hat{P}^\beta_D = y^*$.

5. *Else if* $R(0,x_u) < R(1,y_u)$ *find the base-rate parameter* p_u *such that the optimal isoline of Equation (2.9) intercepts* $ROC^{(\alpha,\beta)}$ *at* (x_u, y_u) *and then set* $\hat{p}^\delta = p_u$, $\hat{P}^\alpha_{FA} = x_u$ *and* $\hat{P}^\beta_D = y_u$. *Otherwise, find the base-rate parameter* p_l *such that the optimal isoline of Equation (2.9) intercepts* $ROC^{(\alpha,\beta)}$ *at* (x_l, y_l) *and then set* $\hat{p}^\delta = p_l$, $\hat{P}^\alpha_{FA} = x_l$ *and* $\hat{P}^\beta_D = y_l$.

The proof of this lemma is very straightforward. The basic idea is that if the uncertainty range of p is large enough, the Nash equilibrium of the game is obtained by selecting the point intercepting equation (3). Otherwise one of the strategies for the intruder is always a dominant strategy of the game and therefore we only need to find which one is it: either $\hat{p} + \delta_u$ or $\hat{p} - \delta_l$. For most practical cases it will be $\hat{p} + \delta_u$. Also note that the optimal operating point in the original *ROC* can be found by obtaining $(\hat{P}_{FA}, \hat{P}_D)$ from $(\hat{P}^\alpha_{FA}, \hat{P}^\beta_D)$.

B. Robust B-ROC Evaluation

Similarly we can now also analyze the robustness of the evaluation done with the B-ROC curves. In this case it is also easy to see that the worst attacker for the evaluation is an intruder I that selects $p_1 = \hat{p} - \delta_l$, $p_2 = \alpha$ and $p_3 = \beta$.

Corollary 1: For any point $(P\hat{P}V, \hat{P}_D)$ *corresponding to* $\hat{p}$ *in the B-ROC curve, a* (δ, α, β) *– intruder can decrease the detection rate and the positive predictive value to the pair* $(P\hat{P}V^{\delta,\alpha,\beta}, \hat{P}^\beta_D)$, *where* $\hat{P}^\beta = \hat{P}_D(1-\beta)$ *and where*

$$P\hat{P}V^{\delta,\alpha,\beta} = \frac{P^\beta_D p - P^\beta \delta}{P^\beta_D p + P^\alpha_{FA}(1-p) + \delta P^\alpha_{FA} - \delta P^\beta_D} \tag{3.6}$$

C. Example: Minimizing the Cost of a Chosen Intrusion Rate Attack

In this example we introduce probably one of the easiest formulations of an attacker against a classifier: we assume that the attacker cannot change its feature vectors **x**, but rather only its frequency of attacks: p. This example also shows the generality of lemma 2 and also presents a compelling scenario of when does a probabilistic IDSs make sense.

Assume an ad hoc network scenario similar to [33, 34, 35, 36] where nodes monitor and distribute reputation values of other nodes' behavior at the routing layer. The monitoring nodes report selfish actions (e.g. nodes that agree to forward packets in order to be accepted in the network, but then fail to do so) or attacks (e.g. nodes that modify routing information before forwarding it).

Now suppose that there is a network operator considering implementing a watchdog monitoring scheme to check the compliance of nodes forwarding packets as in [33]. The operator

then plans an evaluation period of the method where trusted nodes will be the watchdogs reporting the misbehavior of other nodes. Since the detection of misbehaving nodes is not perfect, during the evaluation period the network operator is going to measure the consistency of reports given by several watchdogs and decide if the watchdog system is worth keeping or not.

During this trial period, it is of interest to selfish nodes to behave as deceiving as they can so that the neighboring watchdogs have largely different results and the system is not permanently established. As stated in [33] the watchdogs might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion or 6) partial dropping. False alarms are also possible in several cases, for example when a node moves out of the previous node's listening range before forwarding on a packet. Also if a collision occurs while the watchdog is waiting for the next node to forward a packet, it may never overhear the packet being transmitted.

We now briefly describe this model according to our guidelines:

Desired Properties Assume the operator wants to find a strategy that minimizes the probability of making errors. This is an example of the expected cost metric function with $C(0,0) = C(1,1) = 0$ and $C(1,0) = C(0,1) = 1$.

Feasible Design Space $\mathcal{DM} = \{\pi_i \in [0,1] : \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1\}$.

Information Available to the Adversary We assume the adversary knows everything that we know and can make inferences about the situation the same way as we can. In game theory this adversaries are usually referred to as *intelligent*.

Capabilities of the Adversary The adversary has complete control over the base-rate p (its frequency of attacks). The feasible set is therefore $\mathcal{F} = [0,1]$.

Goal of the Adversary Evaluation attack.

Evaluation Metric

$$r^* = \min_{\pi_i \in \mathcal{DM}} \max_{p \in \mathcal{F}} \mathbb{E}[C(I,A)]$$

Note the order in optimization of the evaluation metric. In this case we are assuming that the operator of the IDS makes the first decision, and that this information is then available to the attacker when selecting the optimal p. We call the strategy of the operator *secure* if the expected cost (probability of error) is never greater than r^* for any feasible adversary.

Model Assumptions We have assumed that the attacker will not be able to change $\hat{P}_{FA}$ and $\hat{P}_D$. This results from its assumed inability to directly modify the feature vector $\mathbf{x}$.

Notice that in this case the detector algorithm $\mathcal{D}$ is the watchdog mechanism that monitors the medium to see if the packet was forwarded F or if it did not hear the packet being forwarded (unheard U) during a specified amount of time. Following [33] (where it is shown that the number of false alarms can be quite high) we assume that a given watchdog $\mathcal{D}$ has a false alarm rate of $\hat{P}_{FA} = 0.5$ and a detection rate of $\hat{P}_D = 0.75$. Given this detector algorithm, a (non-randomized) decision maker $\mathcal{DM}$ has to be one of the following rules (where intuitively, h_3 is the more appealing):

$$\begin{array}{ll} h_1(F) = 0 & h_1(U) = 0 \\ h_2(F) = 1 & h_2(U) = 0 \\ h_3(F) = 0 & h_3(U) = 1 \\ h_4(F) = 1 & h_4(U) = 1 \end{array}$$

	h_0	h_1	h_2	h_3
$I=0$	$R(0,0)$	$R(0,\hat{P}_D)$	$R(0,\hat{P}_{FA})$	$R(0,1)$
$I=1$	$R(1,0)$	$R(1,\hat{P}_{FA})$	$R(1,\hat{P}_D)$	$R(1,1)$

Table 3.1: Matrix for the zero-sum game theoretic formulation of the detection problem

Since the operator wants to check the consistency of the reports, the selfish nodes will try to maximize the probability of error (i.e. $C(0,0) = C(1,1) = 0$ and $C(0,1) = C(1,0) = 1$) of any watchdog with a chosen intrusion rate attack. As stated in lemma 2, this is a zero-sum game where the adversary is the maximizer and the watchdog is the minimizer. The matrix of this game is given in Table 3.1.

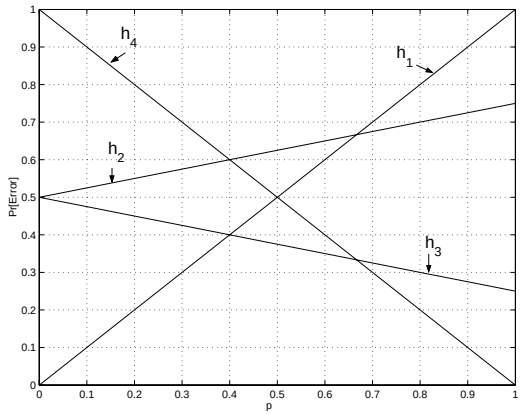


Figure 3.1: Probability of error for h_i vs. p

It is a well known fact that in order to achieve a Nash equilibrium of the game, the players should consider mixed strategies (i.e. consider probabilistic choices). For our example the optimal mixed strategy for the selfish node (see Figure 3.1) is to drop a packet with probability $p^* = \hat{P}_{FA}/(\hat{P}_{FA1} + \hat{P}_D)$. On the other hand the optimal strategy for $\mathcal{DM}$ is to select h_3 with probability $1/(\hat{P}_{FA} + \hat{P}_D)$ and h_1 with probability $(\hat{P}_{FA} - (1 - \hat{P}_D))/(\hat{P}_{FA} - (1 - \hat{P}_D) + 1)$. This example shows that sometimes in order to minimize the probability of error (or any general cost) against an adaptive attacker, $\mathcal{DM}$ has to be a probabilistic algorithm.

Lemma 2 also presents a way to get this optimal point from the ROC, however it is not obvious at the beginning how to get the same results, as there appear to be only three points in the ROC: $(P_{FA} = 0, P_D = 0)$ (by selecting h_1), $(\hat{P}_{FA} = 1/2, \hat{P}_D = 3/4)$ (by selecting h_3) and $(P_{FA} = 1, P_D = 1)$ (by selecting h_4). The key property of ROC curves to remember is that the (optimal) ROC curve is a continuous and concave function [27], and that in fact, the points that do not correspond to deterministic decisions are joined by a straight line whose points can be achieved by a mixture of probabilities of the extreme points. In our case, the line $y = 1 - x$ intercepts the (optimal) ROC at the optimal operating points $\hat{P}^*_{FA} = \hat{P}_{FA}/(\hat{P}_D + \hat{P}_{FA})$ and $\hat{P}^*_D = \hat{P}_D/(\hat{P}_{FA} + \hat{P}_D)$ (see Figure 3.2). Also note that p^* is the value required to make the slope of the isoline parallel to the ROC line intersecting (P^*_{FA}, P^*_D).

The optimal strategy for the intruder is therefore $p^* = 2/5$, while the optimal strategy for $\mathcal{DM}$ is to select h_1 with probability $1/5$ and h_3 with probability $4/5$. In the robust operating point we have $P^*_{FA} = 2/5$ and $P^*_D = 3/5$. Therefore, after fixing $\mathcal{DM}$, it does not matter if p deviates from p^* because we are guaranteed that the probability of error will be no worse (but

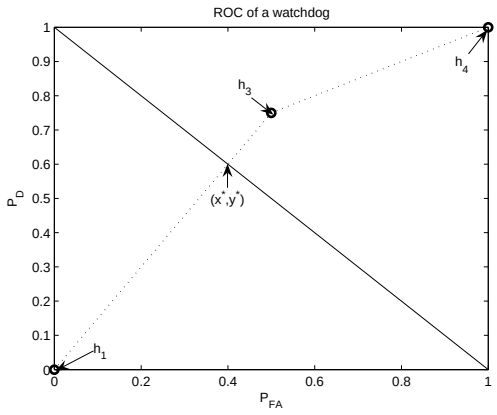


Figure 3.2: The optimal operating point

no better either) than $2/5$, therefore the IDS can be claimed to be $2/5$-robust.

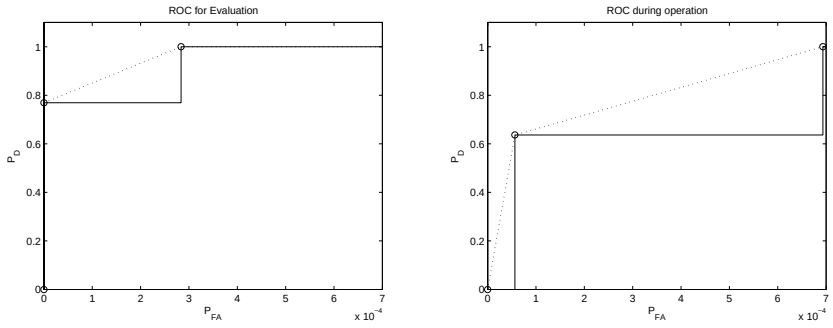
D. Example: Robust Evaluation of IDSs

As a second example, we chose to perform an intrusion detection experiment with the 1998 MIT/Lincoln Labs data set [37]. Although several aspects of this data set have been criticized in [38], we still chose it for two main reasons. On one hand, it has been (and arguably still remains) the most used large-scale data set to evaluate IDSs. In the second place we are not claiming to have a better IDS to detect attacks and then proving our claim with its good performance in the MIT data set (a feat that would require further testing in order to be assured on the quality of the IDS). Our aim on the other hand is to illustrate our methodology, and since this data set is publicly available and has been widely studied and experimented with (researchers can in principle reproduce any result shown in a paper), we believe it provides the basic background and setting to exemplify our approach.

Of interest are the Solaris system log files, known as BSM logs. The first step of the experiment was to record every instance of a program being executed in the data set. Next, we created a very simple tool to perform buffer overflow detection. To this end, we compared the buffer length of each execution with a buffer threshold, if the buffer size of the execution was larger than the threshold we report an alarm.

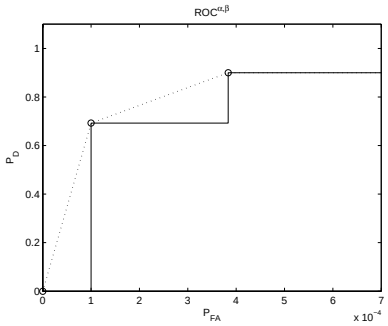
We divided the data set into two sets. In the first one (weeks 6 and 7), our IDS performs very well and thus we assume that this is the ”evaluation” period. The previous three weeks were used as the period of operation of the IDS. Figure 3.3(a)[4] shows the results for the ”evaluation” period when the buffer threshold ranges between 64 and 780. The dotted lines represent the suboptimal points of the ROC or equivalently the optimal points that can be achieved through randomization. For example the dotted line of Figure 3.3(a) can be achieved by selecting with probability λ the detector with threshold 399 and with probability $1-\lambda$ the

[4]Care must always be taken when looking at the results of ROC curves due to the ”unit of analysis” problem [38]. For example comparing the ROC of Figure 3.3(a) with the ROC of [6] one might arrive to the erroneous conclusion that the buffer threshold mechanism produces an IDS that is better than the more sophisticated IDS based on Bayesian networks. The difference lies in the fact that we are monitoring the execution of *every program* while the experiments in [6] only monitor the attacked programs (`eject`, `fbconfig`, `fdformat` and `ps`). Therefore although we raise more false alarms, our false alarm rate (number of false alarms divided by the total number of honest executions) is smaller.



(a) Original ROC obtained during the evaluation period

(b) Effective ROC during operation time



(c) Original ROC under adversarial attack $ROC^{\alpha,\beta}$

Figure 3.3: Robust expected cost evaluation

detector with threshold 773 and letting λ range from zero to one.

During the evaluation weeks there were 81108 executions monitored and 13 attacks, therefore $\hat{p} = 1.6 \times 10^{-4}$. Assuming that our costs (per execution) are $C(0,0) = C(1,1) = 0$, $C(1,0) = 850$ and $C(0,1) = 100$ we find that the slope given by equation 2.8 is $m_{C,\hat{p}} = 735.2$, and therefore the optimal point is $(2.83 \times 10^{-4}, 1)$, which corresponds to a threshold of 399 (i.e. all executions with buffer sizes bigger than 399 raise alarms). Finally, with these operating conditions we find out that the expected cost (per execution) of the IDS is $\mathbb{E}[C(I,A)] = 2.83 \times 10^{-2}$.

In the previous three weeks used as the "operation" period our buffer threshold does not perform as well, as can be seen from its ROC (shown in Figure 3.3(b).) Therefore if we use the point recommended in the evaluation (i.e. the threshold of 399) we get an expected cost of $\mathbb{E}^{\text{operation}}[C(I,A)] = 6.934 \times 10^{-2}$. Notice how larger the expected cost per execution is from the one we had evaluated. This is very noticeable in particular because the base-rate is smaller during the operation period ($\hat{p}^{\text{operation}} = 7 \times 10^{-5}$) and a smaller base-rate should have given us a smaller cost.

To understand the new ROC let us take a closer look at the performance of one of the thresholds. For example, the buffer length of 773 which was able to detect 10 out of the 13 attacks at no false alarm in Figure 3.3(a) does not perform well in Figure 3.3(b) because some programs such as `grep`, `awk`, `find` and `ld` were executed under normal operation with long string lengths. Furthermore, a larger percent of attacks was able to get past this threshold. This is in general the behavior modeled by the parameters α and β that the adversary has access to in our framework.

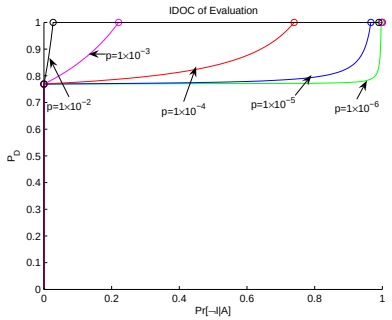
Let us begin the evaluation process from the scratch by assuming a $([1 \times 10^{-5}, 0], 1 \times 10^{-4}, 0.1) - \textit{intruder}$, where $\delta = [1 \times 10^{-5}, 0]$ means the IDS evaluator believes that the base-rate during operation will be at most $\hat{p}$ and at least $\hat{p} - 1 \times 10^{-5}$. $\alpha = 1 \times 10^{-5}$ means that the IDS evaluator believes that new normal behavior will have the chance of firing an alarm with probability 1×10^{-5}. And $\beta = 0.1$ means that the IDS operator has estimated that ten percent of the attacks during operation will go undetected. With these parameters we get the $ROC^{\alpha,\beta}$ shown in Figure 3.3(c).

Note that in this case, p is bounded in such a way that the equilibrium of the game is achieved via a pure strategy. In fact, the optimal strategy of the intruder is to attack with frequency $\hat{p} + \delta_u$ (and of course, generate missed detections with probability β and false alarms with probability α) whereas the optimal strategy of $\mathcal{DM}$ is to find the point in $ROC^{\alpha,\beta}$ that minimizes the expected cost by assuming that the base-rate is $\hat{p} + \delta_u$.

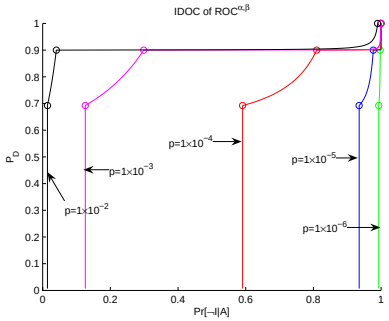
The optimal point for the $ROC^{\alpha,\beta}$ curve corresponds to the one with threshold 799, having an expected cost $\mathbb{E}^{\delta,\alpha,\beta}[C(I,A)] = 5.19 \times 10^{-2}$. Finally, by using the optimal point for $ROC^{\alpha,\beta}$, as opposed to the original one, we get during operation an expected cost of $\mathbb{E}^{\text{operation}}[C(I,A)] = 2.73 \times 10^{-2}$. Therefore in this case, not only we have maintained our expected $5.19 \times 10^{-2} -$ *security* of the evaluation, but in addition the new optimal point actually performed better than the original one.

Notice that the evaluation of Figure 3.3 relates exactly to the problem we presented in the introduction, because it can be thought of as the evaluation of two IDSs. One IDS having a buffer threshold of length 399 and another IDS having a buffer threshold of length 773. Under ideal conditions we choose the IDS of buffer threshold length of 399 since it has a lower expected cost. However after evaluating the worst possible behavior of the IDSs we decide to select the one with buffer threshold length of 773.

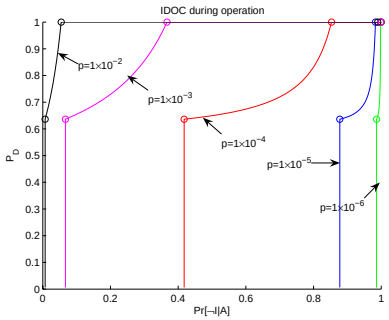
An alternative view can be achieved by the use of B-ROC curves. In Figure 3.4(a) we



(a) Original B-ROC obtained during the evaluation period



(b) B-ROC obtained from $ROC^{\alpha,\beta}$



(c) B-ROC during operation time

Figure 3.4: Robust B-ROC evaluation

see the original B-ROC curve during the evaluation period. These curves give a false sense of confidence in the IDS. Therefore we study the B-ROC curves based on $ROC^{\alpha,\beta}$ in Figure 3.4(b). In Figure 3.4(c) we can see how the B-ROC of the actual operating environment follows more closely the B-ROC based on $ROC^{\alpha,\beta}$ than the original one.

IV A White Box Adversary Model

So far we have been treating the decision making process without considering the exact way that an adversary can modify or create the input to the classifier x. However following the same statistical framework we have been considering there is a way to formulate this problem in a way that sheds light into the exact behavior and optimal adversarial strategies.

In order to define this problem recall first that an *optimal* decision rule (optimal in the sense that it minimizes several evaluation metrics such as the probability of error, the expected cost or the probability of missed positives given an upper bound on the probability of false positives) is the log-likelihood ratio test $\ln \frac{f(x|1)}{f(x|0)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau$, where H_i denotes the hypothesis that $I = i$, and τ depends on the particular evaluation metric and its assumptions (e.g., misclassification costs, base-rates etc.). If the log-likelihood ratio is greater than τ then the detector outputs $A = 1$ and if it less than τ, the detector outputs $A = 0$ (if it is equal to τ the detector randomly selects the hypothesis based on the evaluation metric being considered).

This new detailed formulation allows us to model the fact that an attacker is able to modify its attack strategy $f(x|1)$. The pdf $f(x|0)$ of the normal behavior can be learnt via one-

class learning, however since the attacker can change its strategy, we cannot trust a machine learning technique to estimate $f(x|1)$, since learning a candidate density $f(x|1)$ for an attacker without some proper analysis may result in serious performance degradation if the attacker's strategy diverges from our estimated model.

It is therefore one of our aims in the following two chapters to evaluate and design algorithms that try to estimate the least favorable pdf $f(x|1)$ within a feasible adversary class $\mathcal{F}$. The adversary class $\mathcal{F}$ will consists of pdfs $f(x|1)$ constrained by a requirement that an adversary has to satisfy. For example in the next chapter it is a desired level of wireless channel access, and in chapter **??** the feasible class $\mathcal{F}$ is composed of pdfs that satisfy certain distortion constraints.

An abstract example on how to model the problem of finding the least favorable $f(x|1)$ is now considered. Assume an adversary whose objective is to create mimicry attacks: i.e., it wants to minimize the probability of being detected. Furthermore assume x takes only discrete values, so $f(x|i)$ are in fact probability mass functions (pmfs) (as opposed to density functions). A way to formulate this problem can be done with the help of information theory inequalities [39]:

$$P_{FA}\log\frac{P_{FA}}{P_D}+(1-P_{FA})\log\frac{1-P_{FA}}{1-P_D}\leq KL[f(x|0)||f(x|1)] \tag{3.7}$$

where $KL[f_0||f_1]$ denotes the Kullback-Leibler divergence between two pmfs f_0 and f_1. From this inequality it can be deduced that if we fix P_{FA} to be very small ($P_{FA}\to 0$), then

$$P_D\leq 1-2^{-KL[f(x|0)||f(x|1)]} \tag{3.8}$$

The task of the adversary would be therefore the following:

$$h^*=\arg\min_{h\in\mathcal{F}} KL[f(x|0)||h(x)]$$

It can be shown in fact that some of the examples studied in the next two chapters can be formulated as exactly performing this optimization problem.

Besides estimating least favorable distributions, another basic notion in minimax game approaches that is going to be very useful in the next two sections is that of a *saddle point*. A strategy (detection rule) $d^\star$ and an operating point (attack) $f(x|1)^\star$ in the uncertainty class $\mathcal{F}$ form a saddle point if:

1. For the attack $f(x|1)^\star$, any detection rule d other than $d^\star$ has worse performance. Namely $d^\star$ is the optimal detection rule for attack $f(x|1)^\star$ in terms of minimizing the objective function (evaluation).

2. For the detection rule $d^\star$, any attack $f(x|1)$ from the uncertainty class, other than $f(x|1)^\star$ gives better performance. Namely, detection rule $d^\star$ has its worst performance for attack $f(x|1)^\star$.

Implicit in the minimax approach is the assumption that the attacker has full knowledge of the employed detection rule. Thus, it can create a misbehavior strategy that maximizes the cost of the performance of the classifier. Therefore, our approach refers to the case of an intelligent attacker that can adapt its misbehavior policy so as to avoid detection. One issue that needs to be clarified is the structure of this attack strategy. Subsequently, by deriving the

detection rule and the performance for that case, we can obtain an (attainable) upper bound on performance over all possible attacks.

Even though we did not point it out before, it can in fact be shown that the optimal operating points in the two IDS examples presented section III are in fact saddle point equilibria!

V Conclusions

There are two main problems that any empirical test of a classifier will face. The first problem relates to the inferences that once can make about any classifier system based on experiments alone. An example is the low confidence on the estimate for the probability of detection in the ROC. A typical way to improve this estimate in other classification tasks is through the use of error bars in the ROC. However, if tests of classifiers include very few attacks and their variations, there is not enough data to provide an accurate significance level for the bars. Furthermore, the use of error bars and any other cross-validation technique gives the average performance of the classifier. However, this brings us to the second problem, and it is the fact that since the classifiers are subject to an adversarial environment, evaluating such a decision maker based on its average performance is not enough. Our adversary models tries to address these two problems, since it provides a principled approach to give us the worst case performance of a classifier.

The extent by which the analysis with a $(\delta, \alpha, \beta) - \mathit{intruder}$ will follow the real operation of the IDS will depend on how accurately the person doing the evaluation of the IDS understands the IDS and its environment, for example, to what extent can the IDS be evaded, how well the signatures are written (e.g. how likely is it that normal events fire alarms) etc. However, by assuming *robust* parameters we are actually assuming a pessimistic setting, and if this pessimistic scenario never happens, we might be operating at a suboptimal point (i.e. we might have been too pessimistic in the evaluation).

Finally, although the white box framework for adversarial modeling gives us a fine grained evaluation procedure, it is not always a better alternative to the black box adversary model. There are basically two problems with the white box adversary model. The first problem is the fact that finding the optimal adversarial distribution is usually an intractable problem. The second problem is that sometimes in order to avoid the intractability of the problem, $f(x|1)$ is sometimes assumed to follow a certain parametric model. Therefore instead of searching for optimal adversarial pdfs $f(x|1)$ the problem is replaced with one of finding the optimal parameters of a prescribed distribution. The problem with this approach is that assuming a distribution form creates extra assumptions about the attacker, and as explained in the guidelines defined at the beginning of this chapter, any extra assumption must be taken with care. Specially because in practice the adversary will most likely not follow the parametric distribution assumed a priori.

Chapter 4
Performance Comparison of MAC layer Misbehavior Schemes

If anything could dissipate my love to humanity, it would be ingratitude. In short, I am a hired servant, I expect my payment at once- that is, praise, and the repayment of love with love. Otherwise I am incapable of loving anyone.
- The Brothers Karamazov, Dostoevsky

I Overview

This chapter revisits the problem of detecting greedy behavior in the IEEE 802.11 MAC protocol by evaluating the performance of two previously proposed schemes: DOMINO and the Sequential Probability Ratio Test (SPRT). The evaluation is carried out in four steps. We first derive a new analytical formulation of the SPRT that takes into account the discrete nature of the problem. Then we develop a new tractable analytical model for DOMINO. As a third step, we evaluate the theoretical performance of SPRT and DOMINO with newly introduced metrics that take into account the repeated nature of the tests. This theoretical comparison provides two major insights into the problem: it confirms the optimality of SPRT and motivates us to define yet another test, a nonparametric CUSUM statistic that shares the same intuition as DOMINO but gives better performance. We finalize this chapter with experimental results, confirming the correctness of our theoretical analysis and validating the introduction of the new nonparametric CUSUM statistic.

II Introduction

The problem of deviation from legitimate protocol operation in wireless networks and efficient detection of such behavior has become a significant issue in recent years. In this work we address and quantify the impact of MAC layer attacks that aim at disrupting critical network functionalities and information flow in wireless networks.

It is important to note that parameters used for deriving a decision of whether a protocol participant misbehaves or not should be carefully chosen. For example, choosing the percentage of time the node accesses the channel as a misbehavior metric can result in a high number of false alarms due to the fact that the other protocol participants might not have anything (or have significantly less traffic) to transmit within a given observation period. This could easily lead to false accusations of legitimate nodes that have large amount of data to send. Measuring throughput offers no indication of misbehavior since it is impossible to define “legitimate” throughput. Therefore, it is reasonable to use either fixed protocol parameters (such as SIFS or DIFS window size) or parameters that belong to a certain (fixed) range of values for monitoring misbehavior (such as backoff).

In this work we derive analytical bounds of two previously proposed protocols for detecting random access misbehavior: DOMINO [40] and SPRT-based tests [41, 42] and show the optimality of SPRT against the worst-case adversary for all configurations of DOMINO. Following the main idea of DOMINO, we introduce a nonparametric CUSUM statistic that

shares the same intuition as DOMINO but gives better performance for all configurations of DOMINO.

A. Background Work

MAC layer protocol misbehavior has been studied in various scenarios and mathematical frameworks. The random nature of access protocols coupled with the highly volatile nature of wireless medium poses the major obstacle in generation of the unified framework for misbehavior detection. The goals of a misbehaving peer can range from exploitation of available network resources for its own benefit up to network disruption. An efficient Intrusion Detection System should exhibit a capability to detect a wide range of misbehavior policies with an acceptable False Alarm rate. This presents a major challenge due to the nature of wireless protocols.

The current literature offers two major approaches in the field of misbehavior detection. The first set of approaches provides solutions based on modification of the current IEEE 802.11 MAC layer protocol by making each protocol participant aware of the backoff values of its neighbors. The approach proposed in [43] assumes existence of a trustworthy receiver that can detect misbehavior of the sender and penalize it by assigning him higher back-off values for subsequent transmissions. A decision about protocol deviation is reached if the observed number of idle slots of the sender is smaller than a pre-specified fraction of the allocated back-off. The sender is labeled as misbehaving if it turns out to deviate continuously based on a cumulative metric over a sliding window. The work in [44] attempts to prevent scenarios of colluding sender-receiver pairs by ensuring randomness in the course of the MAC protocol.

A different line of thought is followed in [40, 41, 42], where the authors propose a misbehavior detection scheme without making any changes to the actual protocol. In [40] the authors focus on multiple misbehavior policies in the wireless environment and put emphasis on detection of backoff misbehavior. They propose a sequence of conditions on available observations for testing the extent to which MAC protocol parameters have been manipulated. The proposed scheme does not address the scenarios that include intelligent adaptive cheaters or collaborating misbehaving nodes. The authors in [41, 42] address the detection of an adaptive intelligent attacker by casting the problem of misbehavior detection within the minimax robust detection framework. They optimize the system's performance for the worst-case instance of uncertainty by identifying the least favorable operating point of a system and derive the strategy that optimizes the system's performance when operating in that point. The system performance is measured in terms of number of required observation samples to derive a decision (detection delay).

However, DOMINO and SPRT were presented independently, without direct comparison or performance analysis. Additionally, both approaches evaluate the detection scheme performance under unrealistic conditions, such as probability of false alarm being equal to 0.01, which in our simulations results in roughly 700 false alarms per minute (in saturation conditions), a rate that is unacceptable in any real-life implementation. Our work contributes to the current literature by: (i) deriving a new pmf for the worst case attack using an SPRT-based detection scheme, (ii) providing new performance metrics that address the large number of alarms in the evaluation of previous proposals, (iii) providing a complete analytical model of DOMINO in order to obtain a theoretical comparison to SPRT-based tests and (iv) proposing an improvement to DOMINO based on the CUSUM test.

The rest of the chapter is organized as follows. Sect. III outlines the general setup of the problem. In Sect. IV we propose a minimax robust detection model and derive an expression for the worst-case attack in discrete time. In Sect. V we provide extensive analysis of DOMINO,

followed by the theoretical comparison of two algorithms in Sect. VI. Motivated by the main idea of DOMINO, we offer a simple extension to the algorithm that significantly improves its performance in Sect. VII. In Sect. VIII we present the experimental performance comparison of all algorithms. Finally, Sect. IX concludes our study. In subsequent sections, the terms "attacker" and "adversary" will be used interchangeably with the same meaning.

III Problem Description and Assumptions

Throughout this work we assume existence of an intelligent adaptive attacker that is aware of the environment and its changes over a given period of time. Consequently, the attacker is able to adjust its access strategy depending on the level of congestion in its environment. Namely, we assume that, in order to minimize the probability of detection, the attacker chooses legitimate over selfish behavior when the level of congestion in the network is low. Similarly, the attacker chooses adaptive selfish strategy in congested environments. Due to the previously mentioned reasons, we assume a benchmark scenario where all the participants are backlogged, i.e., have packets to send at any given time in both theoretical and experimental evaluations. We assume that the attacker will employ the worst-case misbehavior strategy in this setting, and consequently the detection system can estimate the maximal detection delay. It is important to mention that this setting represents the worst-case scenario with regard to the number of false alarms per unit of time due to the fact that the detection system is forced to make maximum number of decisions per time unit.

In order to characterize the strategy of an intelligent attacker, we assume that both misbehaving and legitimate node attempt to access the channel simultaneously. Consequently, each station generates a sequence of random backoffs $X_1, X_2, \ldots, X_i$ over a fixed period of time. Accordingly, the backoff values, $X_1, X_2, \ldots, X_i$, of each legitimate protocol participant are distributed according to the probability mass function (pmf) $p_0(x_1, x_2, \ldots, x_i)$. The pmf of the misbehaving participants is unknown to the system and is denoted with $p_1(x_1, x_2, \ldots, x_i)$, where $X_1, X_2, \ldots, X_i$ represent the sequence of backoff values generated by the misbehaving node over the same period of time.

The assumption that holds throughout this chapter is that a detection agent (e.g., the access point) monitors and collects the backoff values of a given station. It is important to note that observations are not perfect and can be hindered by concurrent transmissions or external sources of noise. It is impossible for a passive monitoring agent to know the backoff stage of a given monitored station due to collisions and to the fact that in practice, nodes might not be constantly backlogged. Furthermore, in practical applications the number of false alarms in anomaly detection schemes is very high. Consequently, instead of building a "normal" profile of network operation with anomaly detection schemes, we utilize specification based detection. In our setup we identify "normal" (i.e., a behavior consistent with the 802.11 specification) profile of a backlogged station in the IEEE 802.11 without any competing nodes, and notice that its backoff process $X_1, X_2, \ldots, X_i$ can be characterized with pdf $p_0(x_i) = 1/(W+1)$ for $x_i \in \{0, 1, \ldots, W\}$ and zero otherwise. We claim that this assumption minimizes the probability of false alarms due to imperfect observations. At the same time, a safe upper bound on the amount of damaging effects a misbehaving station can cause to the network is maintained.

Although our theoretical results utilize the above expression for p_0, the experimental setting utilizes the original implementation of the IEEE 802.11 MAC. In this case, the detection agent needs to deal with observed values of x_i larger than W, which can be due to collisions or due to the exponential backoff specification in the IEEE 802.11. We further discuss this issue

in Sect. VIII.

IV Sequential Probability Ratio Test (SPRT)

Due to the nature of the IEEE 802.11 MAC, the back-off measurements are enhanced by an additional sample each time a node attempts to access the channel. Intuitively, this gives rise to the employment of a sequential detection scheme in the observed problem. The objective of the detection test is to derive a decision as to whether or not misbehavior occurs with the least number of observations. A sequential detection test is therefore a procedure which decides whether or not to receive more samples with every new information it obtains. If sufficient information for deriving a decision has been made (i.e. the desired levels of the probability of false alarm and probability of miss are satisfied), the test proceeds to the phase of making a decision.

It is now clear that two quantities are involved in decision making: a stopping time N and a decision rule d_N which at the time of stopping decides between hypotheses H_0 (legitimate behavior) and H_1 (misbehavior). We denote the above combination with D=(N, d_N).

In order to proceed with our analysis we first define the properties of an efficient detector. Intuitively, the starting point in defining a detector should be minimization of the probability of false alarms $\mathbb{P}_0[d_N = 1]$. Additionally, each detector should be able to derive the decision as soon as possible (minimize the number of samples it collects from a misbehaving station) before calling the decision function $\mathbb{E}_1[N]$. Finally, it is also necessary to minimize the probability of deciding that a misbehaving node is acting normally $\mathbb{P}_1[d_N = 0]$. It is now easy to observe that $\mathbb{E}_1[N]$, $\mathbb{P}_0[d_N = 1]$, $\mathbb{P}_1[d_N = 0]$ form a multi-criteria optimization problem. However, not all of the above quantities can be optimized at the same time. Therefore, a natural approach is to define the accuracy of each decision a priori and minimize the number of samples collected:

$$\inf_{D \in \mathcal{T}_{a,b}} \quad \mathbb{E}_1[N] \tag{4.1}$$

where

$$\mathcal{T}_{a,b} = \{(N, d_N) : \mathbb{P}_0[d_N = 1] \leq a \text{ and } \mathbb{P}_1[d_N = 0] \leq b\}$$

The solution D^* (optimality is assured when the data is i.i.d. in both classes) to the above problem is the SPRT [41] with:

$$S_n = \ln \frac{p_1(x_1, \ldots, x_n)}{p_0(x_1, \ldots, x_n)} \text{ and } N = \inf_n S_n \in [L, U].$$

The decision rule d_N is defined as:

$$d_N \quad = \quad \begin{cases} 1 & \text{if } S_N \geq U \\ 0 & \text{if } S_N \leq L, \end{cases} \tag{4.2}$$

where $L \approx \ln \frac{b}{1-a}$ and $U \approx \ln \frac{1-b}{a}$. Furthermore, by Wald's identity:

$$\mathbb{E}_j[N] = \frac{\mathbb{E}_j[S_N]}{\mathbb{E}_j\left[\ln \frac{p_1(x)}{p_0(x)}\right]} = \frac{\mathbb{E}_j[S_N]}{\sum_{x=0}^{W} p_j(x) \ln \frac{p_1(x)}{p_0(x)}} \tag{4.3}$$

with $\mathbb{E}_1[S_N] = Lb + U(1-b)$ and $\mathbb{E}_0[S_N] = L(1-a) + Ua$. We note that the coefficients $j = 0, 1$ in Eq.(4.3) correspond to legitimate and adversarial behavior respectively.

A. Adversary Model

To our knowledge, the current literature does not address the discrete nature of the misbehavior detection problem. This section sets a theoretical framework of the problem in discrete time. Due to the different nature of the problem, the relations derived in [41, 42] no longer hold and a new pmf p_1^* that maximizes the performance of the adversary is derived. We assume the adversary has full control over the probability mass function p_1 and the backoff values it generates. In addition to that we assume that the adversary is intelligent, i. e. he knows everything the detection agent knows and can infer the same conclusions as the detection agent.

Goal of the Adversary: we assume the objective of the adversary is to design an access policy with the resulting probability of channel access P_A, while minimizing the probability of detection. As it has already been mentioned, the optimal access policy results in generation of backoff sequences according to the pmf $p_1^*(x)$.

Theorem 1: The probability that the adversary accesses the channel before any other terminal when competing with n neighboring (honest) terminals for channel access in saturation condition is:

$$\Pr[Access] \equiv P_A = \frac{1}{1 + n\frac{\mathbb{E}_1[X]}{\mathbb{E}_0[X]}} \tag{4.4}$$

Note that when $\mathbb{E}_1[X] = \mathbb{E}_0[X]$ the probability of access is equal for all $n+1$ competing nodes (including the adversary), i.e., all of them will have access probability equal to $\frac{1}{n+1}$. We omit the proof of this theorem and refer the reader to [42] for the detailed derivation.

Solving the above equation for $\mathbb{E}_1[X]$ gives us a constraint on p_1. That is, p_1 must satisfy the following equation:

$$\mathbb{E}_1[X] = \mathbb{E}_0[X]\frac{1 - P_A}{nP_A} \tag{4.5}$$

We now let $g = \frac{1-P_A}{nP_A}$ in order to be able to parametrize the adversary by the scalar g, which intuitively denotes the level of misbehavior by the adversary. For $P_A \in \left\{\frac{1}{1+n}, 1\right\}$, $g \in \{0, 1\}$. Therefore, $g = 0$ and $g = 1$ correspond to legitimate behavior and complete misbehavior respectively. Now, for any given g, p_1 must belong to the class of allowed probability mass functions $\mathcal{A}_g$, where

$$\mathcal{A}_g \equiv \left\{ q : \sum_{x=0}^{W} q(x) = 1 \text{ and } \sum_{x=0}^{W} xq(x) = g\mathbb{E}_0[X] \right\} \tag{4.6}$$

After defining its desired access probability P_A (or equivalently g), the second objective of the attacker is to maximize the amount of time it can misbehave without being detected. Assuming that the adversary has full knowledge of the employed detection test, it attempts to find the access strategy (with pmf p_1) that maximizes the expected duration of misbehavior before an alarm is fired. By looking at equation Eq.(4.3), the attacker thus needs to minimize the following objective function

$$\min_{p_1 \in \mathcal{A}_g} \sum_{x=0}^{W} p_1(x) \ln \frac{p_1(x)}{p_0(x)} \tag{4.7}$$

Theorem 2: Let $g \in \{0, 1\}$ denote the objective of the adversary. The pmf p_1^* that mini-

mizes Eq.(4.7) can be expressed as:

$$p_1^*(x) = \begin{cases} \frac{r^x(r^{-1}-1)}{r^{-1}-r^W} & \text{for } x \in \{0,1,\ldots,W\} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where r is the solution to the following equation:

$$\frac{Wr^W - r^{-1}(Wr^W + r^W - 1)}{(r^{-1}-1)(r^{-1}-r^W)} = g\frac{W}{2} \tag{4.9}$$

Proof: Notice first that the objective function is convex in p_1. We let $q^\varepsilon(x) = p_1^*(x) + \varepsilon h(x)$ and construct the Lagrangian of the objective function and the constraints

$$\sum_{x=0}^{W} q^\varepsilon(x) \ln \frac{q^\varepsilon(x)}{p_0(x)} + \mu_1(\sum_{x=0}^{W} q^\varepsilon(x) - 1) + \mu_2(\sum_{x=0}^{W} x q^\varepsilon(x) - g\mathbb{E}_0[X]) \tag{4.10}$$

By taking the derivative with respect to ε and equating this quantity to zero for all possible sequences $h(x)$, we find that the optimal p_1^* has to be of the form:

$$p_1^*(x) = p_0(x)e^{-\mu_2 x - \mu_0} \tag{4.11}$$

where $\mu_0 = \mu_1 + 1$. In order to obtain the values of the Lagrange multipliers μ_0 and μ_2 we utilize the fact that $p_0(x) = \frac{1}{W+1}$. Additionally, we utilize the constraints in $\mathcal{A}_g$. The first constraint states that p_1^* must be a pmf and therefore by setting Eq.(4.11) equal to one and solving for μ_0 we have

$$\mu_0 = \ln \sum_{x=0}^{W} p_0(x) r^x = \ln \frac{1}{W+1} \frac{r - r^W}{r-1} \tag{4.12}$$

where $r = e^{-\mu_2}$. Replacing this solution in Eq. 4.11 we get

$$p_1^*(x) = \frac{r^x(r^{-1}-1)}{r^{-1}-r^W} \tag{4.13}$$

The second constraint in $\mathcal{A}_g$ is rewritten in terms of Eq.(4.13) as

$$\frac{r^{-1}-1}{r^{-1}-r^W} \sum_{x=0}^{W} x r^x = g\mathbb{E}_0[X] \tag{4.14}$$

from where Eq.(4.9) follows. ■

Fig. 4.1 illustrates the optimal distribution p_1^* for two values of the parameter g.

B. SPRT Optimality for any Adversary in $\mathcal{A}_g$

Let $\Phi(D, p_1) = \mathbb{E}_1[N]$. We notice that the above solution was obtained in the form

$$\max_{p_1 \in \mathcal{A}_g} \min_{D \in \mathcal{T}_{a,b}} \Phi(D, p_1) \tag{4.15}$$

That is, we first minimized $\Phi(D, p_1)$ with the SPRT (minimization for any p_1) and then found the p_1^* that maximized $\Phi(\text{SPRT}, p_1^*)$.

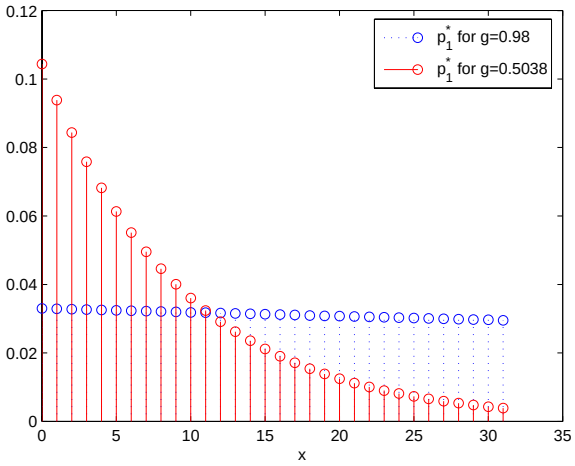


Figure 4.1: Form of the least favorable pmf p_1^* for two different values of g. When g approaches 1, p_1^* approaches p_0. As g decreases, more mass of p_1^* concentrated towards the smaller backoff values.

However, an optimal detector needs to minimize all losses due to the worst-case attacker. That is, the optimal test should in principle be obtained by the following optimization problem

$$\min_{D \in \mathcal{T}_{a,b}} \max_{p_1 \in \mathcal{A}_g} \Phi(D, p_1) \tag{4.16}$$

Fortunately, our solution also satisfies this optimization problem since it forms a saddle point equilibrium, resulting in the following theorem:

Theorem 3: For every $D \in \mathcal{T}_{a,b}$ and every $p_1 \in \mathcal{A}_g$

$$\Phi(D^*, p_1) \leq \Phi(D^*, p_1^*) \leq \Phi(D, p_1^*) \tag{4.17}$$

We omit the proof of the theorem since its derivation follows reasoning similar to the one in [42]. As a consequence of this theorem, no incentive for deviation from (D^*, p_1^*) for any of the players (the detection agent or the misbehaving node) is offered.

C. Evaluation of Repeated SPRT

The original setup of SPRT-based misbehavior detection proposed in [41] was better suited for on-demand monitoring of suspicious nodes (e.g., when a higher layer monitoring agent requests the SPRT to monitor a given node because it is behaving suspiciously, and once it reaches a decision it stops monitoring) and was not implemented as a repeated test.

On the other hand, the configuration of DOMINO is suited for continuous monitoring of neighboring nodes. In order to obtain fair performance comparison of both tests, a repeated SPRT algorithm is implemented: whenever $d_N = 0$, the SPRT restarts with $S_0 = 0$. This setup allows a detection agent to detect misbehavior for both short and long-term attacks. The major problem that arises from this setup is that continuous monitoring can raise a large number of false alarms if the parameters of the test are not chosen appropriately.

This section proposes a new evaluation metric for continuous monitoring of misbehaving nodes. We believe that the performance of the detection algorithms is appropriately captured by employing the expected time before detection $\mathbb{E}[T_D]$ and the average time between false alarms $\mathbb{E}[T_{FA}]$ as the evaluation parameters.

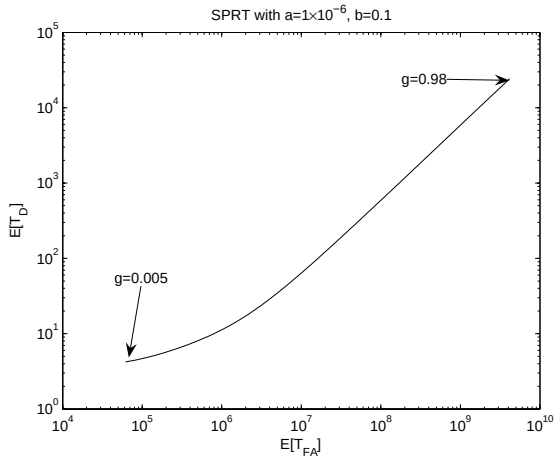


Figure 4.2: Tradeoff curve between the expected number of samples for a false alarm $E[T_{FA}]$ and the expected number of samples for detection $E[T_D]$. For fixed a and b, as g increases (low intensity of the attack) the time to detection or to false alarms increases exponentially.

The above quantities are straightforward to compute for SPRT. Namely, each time the SPRT stops the decision function can be modeled as a Bernoulli trial with parameters a and $1-b$; the waiting time until the first success is then a geometric random variable. Therefore:

$$\mathbb{E}[T_{FA}] = \frac{\mathbb{E}_0[N]}{a} \text{ and } \mathbb{E}[T_D] = \frac{\mathbb{E}_1[N]}{1-b} \tag{4.18}$$

Fig. 4.2 illustrates the tradeoff between these variables for different values of the parameter g. It is important to note that the chosen values of the parameter a in Fig. 4.2 are small. We claim that this represents an accurate estimate of the false alarm rates that need to be satisfied in actual anomaly detection systems [29, 7], a fact that was not taken into account in the evaluation of previously proposed systems.

V Performance analysis of DOMINO

We now present the general outline of the DOMINO detection algorithm. The first step of the algorithm is based on computation of the average value of backoff observations: $X_{ac} = \sum_{i=1}^{m} X_i/m$. In the next step, the averaged value is compared to the given reference backoff value: $X_{ac} < \gamma B$, where the parameter γ $(0 < \gamma < 1)$ is a threshold that controls the tradeoff between the false alarm rate and missed detections. The algorithm utilizes the variable `cheat_count` which stores the number of times the average backoff exceeds the threshold γB. DOMINO raises a false alarm after the threshold is exceeded more than K times. A forgetting factor is considered for `cheat_count` if the monitored station behaves normally in the next monitoring period. That is, the node is partially forgiven: `cheat_count=cheat_count-1` (as long as `cheat_count` remains greater than zero).

We now present the actual detection algorithm from [40]. The algorithm is initialized with `cheat_count` = 0 and after collecting m samples, the following detection algorithm is executed, where `condition` is defined as $\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B$

---

if condition

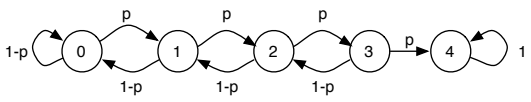


Figure 4.3: For K=3, the state of the variable cheat_count can be represented as a Markov chain with five states. When cheat_count reaches the final state (4 in this case) DOMINO raises an alarm.

```
    cheat_count = cheat_count + 1
    if cheat_count > K
        raise alarm
    end
elseif cheat_count > 0
    cheat_count = cheat_count − 1
end
```

It is now easy to observe that DOMINO is a sequential test, with $N = m * N_t$, where N_t represents the number of steps `cheat_count` takes to exceed K and $d_N = 1$ every time the test stops. We evaluate DOMINO and SPRT with the same performance metrics. However, unlike SPRT where a controls the number of false alarms and b controls the detection rate, the parameters m, γ and K in DOMINO are difficult to tune because there has not been any analysis of their performance. The correlation between DOMINO and SPRT parameters is further addressed in Sec. VIII.

In order to provide an analytical model for the performance of the algorithm, we model the detection mechanism in two steps:

1. We first define $p := \Pr\left[\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B\right]$
2. We define a Markov chain with transition probabilities p and $1-p$. The absorbing state represents the case when misbehavior is detected (note that we assume m is fixed, so p does not depend on the number of observed backoff values). A Markov chain for $K = 3$ is shown in Fig. 4.3.

We can now write

$$p = p^j = \mathbb{P}_j\left[\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B\right], j \in 0,1$$

where $j = 0$ corresponds to the scenario where the samples X_i are generated by a legitimate station $p_0(x)$ and $j = 1$ corresponds to the samples being generated by $p_1^*(x)$. In the remainder of this section we assume $B = \mathbb{E}_0[X_i] = \frac{W}{2}$.

We now derive the expression for p for the case of a legitimate monitored node. Following the reasoning from Sect. III, we assume that each X_i is uniformly distributed on $\{0,1,\ldots,W\}$. It is important to note that this analysis provides a lower bound on the probability of false alarms when the minimum contention window of size $W+1$ is assumed. Using the definition of p we derive the following expression:

$$
\begin{aligned}
p &= \mathbb{P}_0\left[\sum_{i=1}^{m} X_i \leq m\gamma B\right] \\
&= \sum_{k=0}^{\lfloor m\gamma B\rfloor} \mathbb{P}_0\left[\sum_{i=1}^{m} X_i = k\right] \\
&= \sum_{k=0}^{\lfloor m\gamma B\rfloor} \sum_{\{(x_1,\ldots,x_m):\sum_{i=1}^{m} x_i = k\}} \frac{1}{(W+1)^m}
\end{aligned}
\tag{4.19}
$$

where the last equality follows from the fact that the $X_i's$ are i. i. d with pmf $p_0(x_i) = \frac{1}{W+1}$ for all $x_i \in \{0,1,\ldots W\}$.

The number of ways that m integers can sum up to k is $\left(\begin{array}{c} m+k-1 \\ k \end{array}\right)$ and $\sum_{k=0}^{L}\left(\begin{array}{c} m+k-1 \\ k \end{array}\right) = \left(\begin{array}{c} m+L \\ L \end{array}\right)$. An additional constraint is imposed by the fact that X_i can only take values up to W, which is in general smaller than k, and thus the above combinatorial formula cannot be applied. Furthermore, a direct computation of the number of ways x_i bounded integers sum up to k is very expensive. As an example, let $W+1 = 32 = 2^5$ and $m = 10$. A direct summation needed for calculation of p yields at least 2^{50} iterations.

Fortunately, an efficient alternative way for computing $\mathbb{P}_0\left[\sum_{i=1}^{m} X_i = k\right]$ exists. We first define $Y := \sum_{i=1}^{m} X_i$. It is well known that the moment generating function of Y, $M_Y(s) = M_X(s)^m$ can be computed as follows:

$$
\begin{aligned}
M_Y(s) &= \frac{1}{(W+1)^m}\left(1+e^s+\cdots+e^W\right)^m \\
&= \frac{1}{(W+1)^m} \times \\
&\qquad \sum_{\left\{\begin{array}{c} k_0,\ldots,k_W: \\ \sum k_i = m \end{array}\right\}} \left(\begin{array}{c} m \\ k_0;\cdots;k_W \end{array}\right) 1^{k_0} e^{sk_1}\cdots e^{sWk_W}
\end{aligned}
$$

where $\left(\begin{array}{c} m \\ k_0;k_2;\cdots;k_W \end{array}\right)$ is the multinomial coefficient.

By comparing terms with the transform of $M_Y(s)$ we observe that $\Pr[Y=k]$ is the coefficient that corresponds to the term e^{ks} in Eq.(4.20). This result can be used for the efficient computation of p by using Eq.(4.19).

Alternatively, we can approximate the computation of p for large values of m. The approximation arises from the fact that as m increases, Y converges to a Gaussian random variable, by the Central Limit Theorem. Thus,

$$
p = \Pr\left[Y \leq m\gamma B\right] \approx \Phi(z)
$$

where

$$
z = \frac{m\gamma B - m\frac{W}{2}}{\sqrt{(W)(W+2)m/12}}
$$

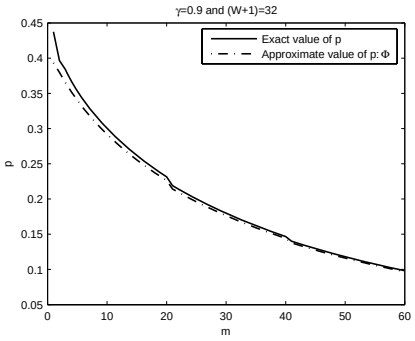


Figure 4.4: Exact and approximate values of p as a function of m.

and $\Phi(z)$ is the error function. Fig. 4.4 illustrates the exact and approximate calculation of p as a function of m, for $\gamma = 0.9$ and $W+1 = 32$. This shows the accuracy of the above approximation for both small and large values of m.

The computation of $p = p^1$ follows the same steps (although the moment generating function cannot be easily expressed in analytical form, it is still computationally tractable) and is therefore omitted.

A. Expected Time to Absorption in the Markov Chain

We now derive the expression for expected time to absorption for a Markov Chain with $K+1$ states. Let μ_i be the expected number of transitions until absorption, given that the process starts at state i. In order to compute the stopping times $\mathbb{E}[T_D]$ and $\mathbb{E}[T_{FA}]$, it is necessary to find the expected time to absorption starting from state zero, μ_0. Therefore, $\mathbb{E}[T_D] = m \times \mu_0$ (computed under $p = p^1$) and $\mathbb{E}[T_{FA}] = m \times \mu_0$ (computed under $p = p^0$).

The expected times to absorption, $\mu_0, \mu_1, \ldots, \mu_{K+1}$ represent the unique solutions of the equations

$$\begin{aligned} \mu_{K+1} &= 0 \\ \mu_i &= 1 + \sum_{j=0}^{K+1} p_{ij}\mu_j \text{ for } i \in \{0, 1 \ldots, K\} \end{aligned}$$

where p_{ij} is the transition probability from state i to state j. For any K, the equations can be represented in matrix form:

$$\begin{bmatrix} -p & p & 0 & \cdots & 0 \\ 1-p & -1 & p & 0 & 0 \\ 0 & 1-p & -1 & p & 0 \\ & & \vdots & & \\ 0 & \cdots & 0 & 1-p & -1 \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_K \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$$

For example, solving the above equations for μ_0 with $K = 3$, the following expression is derived

$$\mu_0 = \frac{1 - p + 2p^2 + 2p^3}{p^4}$$

The expression for μ_0 for any other value of K is obtained in similar fashion.

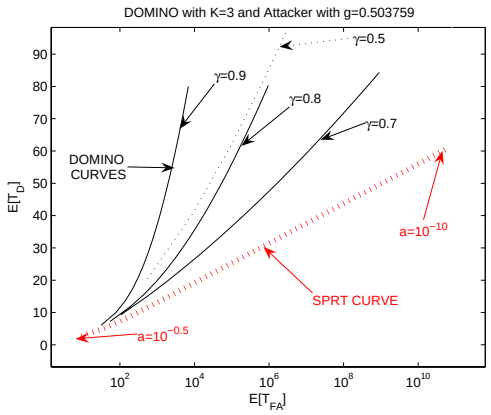


Figure 4.5: DOMINO performance for $K = 3$, m ranges from 1 to 60. γ is shown explicitly. As γ tends to either 0 or 1, the performance of DOMINO decreases. The SPRT outperforms DOMINO regardless of γ and m.

VI Theoretical Comparison

In this section we compare the tradeoff curves between $\mathbb{E}[T_D]$ and $\mathbb{E}[T_{FA}]$ for both algorithms. For the sake of concreteness we compare both algorithms for an attacker with $g = 0.5$. Similar results were observed for other values of g.

For SPRT we set $b = 0.1$ arbitrarily and vary a from $10^{-1/2}$ up to 10^{-10} (motivated by the realistic low false alarm rate required by actual intrusion detection systems [29]). Due to the fact that in DOMINO it is not clear how the parameters m, K and γ affect our metrics, we vary all the available parameters in order to obtain a fair comparison. Fig. 4.5 illustrates the performance of DOMINO for $K = 3$ (the default threshold used in [40]). Each curve for γ has m ranging between 1 and 60, where m represents the number of samples needed for reaching a decision. Observing the results in Fig. 4.5, it is easy to conclude that the best performance of DOMINO is obtained for $\gamma = 0.7$, regardless of m. Therefore, this value of γ is adopted as an optimal threshold in further experiments in order to obtain fair comparison of the two algorithms.

Fig. 4.6 represents the evaluation of DOMINO for $\gamma = 0.7$ with varying threshold K. For each value of K, m ranges from 1 to 60. In this figure, however, we notice that with the increase of K, the point with $m = 1$ forms a performance curve that is better than any other point with $m > 1$.

Consequently, Fig. 4.7 represents the best possible performance for DOMINO; that is, we let $m = 1$ and change K from one up to one hundred. We again test different γ values for this configuration, and conclude that the best γ is still close to the optimal value of 0.7 derived from experiments in Fig. 4.5. However, even with the optimal setting, DOMINO is outperformed by the SPRT.

VII Nonparametric CUSUM statistic

As concluded in the previous section, DOMINO exhibits suboptimal performance for every possible configuration of its parameters. However, the original idea of DOMINO is very intuitive and simple; it compares the observed backoff of the monitored nodes with the expected

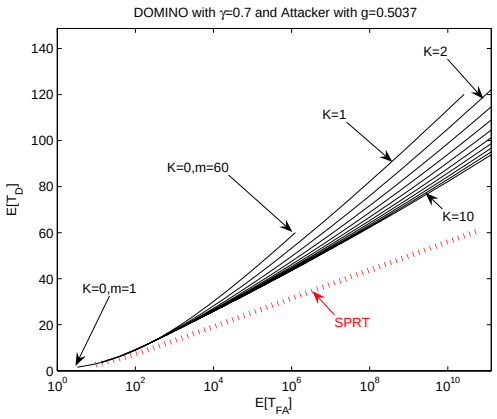


Figure 4.6: DOMINO performance for various thresholds K, $\gamma = 0.7$ and m in the range from 1 to 60. The performance of DOMINO decreases with increase of m. For fixed γ, the SPRT outperforms DOMINO for all values of parameters K and m.

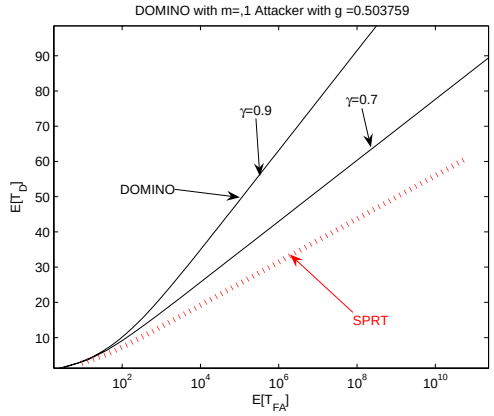


Figure 4.7: The best possible performance of DOMINO is when $m = 1$ and K changes in order to accommodate for the desired level of false alarms. The best γ must be chosen independently.

backoff of honest nodes within a given period of time. This section extends this idea to create a test that we believe will have better performance than DOMINO, while still preserving its simplicity.

Inspired by the notion of nonparametric statistics for change detection, we adapt the nonparametric cumulative sum (CUSUM) statistic and apply it in our analysis. Nonparametric CUSUM is initialized with $y_0 = 0$ and updates its value as follows:

$$y_i = (y_{i-1} - x_i + \gamma B)^+ \tag{4.20}$$

The alarm is fired whenever $y_i > c$.

Assuming $\mathbb{E}_0[X] > \gamma B$ and $\mathbb{E}_1[X] < \gamma B$ (i. e. the expected backoff value of an honest node is always larger than a given threshold and vice versa), the properties of the CUSUM test with regard to the expected false alarm and detection times can be captured by the following theorem.

Theorem 4: The probability of firing a false alarm decreases exponentially with c. Formally, as $c \to \infty$

$$\sup_i |\ln(\mathbb{P}_0[Y_i > c])| = O(c) \tag{4.21}$$

Furthermore, the delay in detection increases only linearly with c. Formally, as $c \to \infty$

$$T_D = \frac{c}{\gamma B - \mathbb{E}_1[X]} \tag{4.22}$$

The proof is a straightforward extension of the case originally considered in [45].

It is easy to observe that the CUSUM test is similar to DOMINO, with c being equivalent to the upper threshold K in DOMINO and the statistic y in CUSUM being equivalent to the variable `cheat_count` in DOMINO when $m = 1$.

The main difference between DOMINO when $m = 1$ and the CUSUM statistic is that every time there is a "suspicious event" (i.e., whenever $x_i \leq \gamma B$), `cheat_count` is increased by one, whereas in CUSUM y_i is increased by an amount proportional to the level of suspected misbehavior. Similarly, when $x_i > \gamma B$, `cheat_count` is decreased only by one (or maintained as zero), while the decrease in y_i can be expressed as $\gamma B - x_i$ (or a decrease of y_i if $y_i - \gamma B - x_i < 0$).

VIII Experimental Results

We now proceed to experimental evaluation of the analyzed detection schemes. It has already been mentioned that we assume existence of an intelligent adaptive attacker that is able to adjust its access strategy depending on the level of congestion in the environment. Namely, we assume that, in order to minimize the probability of detection, the attacker chooses legitimate over selfish behavior when the congestion level is low. Consequently, he chooses adaptive selfish strategy in congested environments. Due to the above reasons, we assume the scenario where all the participants are backlogged, i.e., have packets to send at any given time in constructing the experiments. We assume that the attacker will employ the worst-case misbehavior strategy in this setting, enabling the detection system to estimate the maximal detection delay. It is important to mention that this setting also represents the worst-case scenario with regard to the number of false alarms per unit of time due to the fact that the detection system is forced to make maximum number of decisions per unit of time. We expect the number of alarms to be smaller in practice.

The backoff distribution of an optimal attacker was implemented in the network simulator Opnet and tests were performed for various levels of false alarms. We note that the simulations were performed with nodes that followed the standard IEEE 802.11 access protocol (with exponential backoff). The results presented in this work correspond to the scenario consisting of two legitimate and one selfish node competing for channel access. The detection agent was implemented such that any backoff value $X_i > W$ was set up to be W. We know this is an arbitrary approximation, but our experiments show that it works well in practice. It is important to mention that the resulting performance comparison of DOMINO, CUSUM and SPRT does not change for any number of competing nodes, SPRT always exhibits the best performance. In order to demonstrate the performance of all detection schemes for more aggressive attacks, we choose to present the results for the scenario where the attacker attempts to access channel for 60% of the time (as opposed to 33% if it was behaving legitimately). The backlogged environment in Opnet was created by employing a relatively high packet arrival rate per unit of time: the results were collected for the exponential(0.01) packet arrival rate and the packet size was 2048 bytes. The results for both legitimate and malicious behavior were collected over a fixed period of 100s.

In order to obtain fair performance comparison, a performance metric different from the one in [42] was adopted. The evaluation was performed as a tradeoff between the average time to detection and the average time to false alarm. It is important to mention that the theoretical performance evaluation of both DOMINO and SPRT was measured in number of samples. Here, however, we take advantage of the experimental setup and measure time in number of seconds, a quantity that is more meaningful and intuitive in practice.

We now proceed to the experimental performance analysis of SPRT, CUSUM and DOMINO-based detection schemes. Fig. 4.8 represents the first step in our evaluation. We evaluated the performance of the SPRT using the same parameters as in the theoretical analysis in Sect. VI. DOMINO was evaluated for fixed $\gamma = 0.9$, which corresponds to the value used in the experimental evaluation in [40]. In order to compare the performance to SPRT, we vary the value of K, which essentially determines the number of false alarms. We observe the performance of DOMINO for 2 different values of parameter m. As it can be seen from Fig 4.8, SPRT outperforms DOMINO for all values of K and m. We note that the best performance of DOMINO was obtained for $m = 1$ (the detection delay is smaller when the decision is made after every sample). Therefore, we adopt $m = 1$ for further analysis of DOMINO. Fig. 4.9 reproduces the setting used for theoretical analysis in Fig. 4.7. Naturally, we obtain the same results as in Fig. 4.7 and choose $\gamma = 0.7$ for the final performance analysis of DOMINO.

After finding the optimal values of γ and m we now perform final evaluation of DOMINO, CUSUM and SPRT. The results are presented in Fig. 4.10. We observe that even for the optimal setting of DOMINO, the SPRT outperforms it for all values of K. We also note that due to the reasons explained in Sect. VII, the CUSUM test experiences detection delays similar to the ones of the SPRT.

If the logarithmic x-axis in the tradeoff curves in Sect. VI is replaced with a linear one, we can better appreciate how accurately our theoretical results match the experimental evidence (Fig. 4.11).

IX Conclusions and future work

In this work, we performed extensive analytical and experimental comparison of the existing misbehavior detection schemes in the IEEE 802.11 MAC. We confirm the optimality of

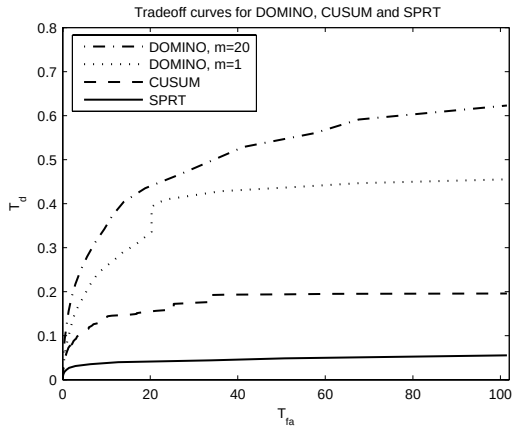


Figure 4.8: Tradeoff curves for each of the proposed algorithms. DOMINO has parameters $\gamma = 0.9$ and $m = 1$ while K is the variable parameter. The nonparametric CUSUM algorithm has as variable parameter c and the SPRT has $b = 0.1$ and a is the variable parameter.

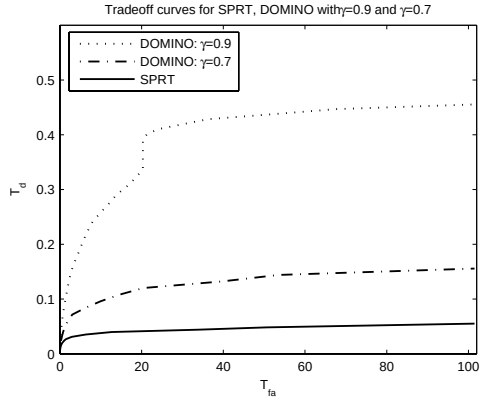


Figure 4.9: Tradeoff curves for default DOMINO configuration with $\gamma = 0.9$, best performing DOMINO configuration with $\gamma = 0.7$ and SPRT.

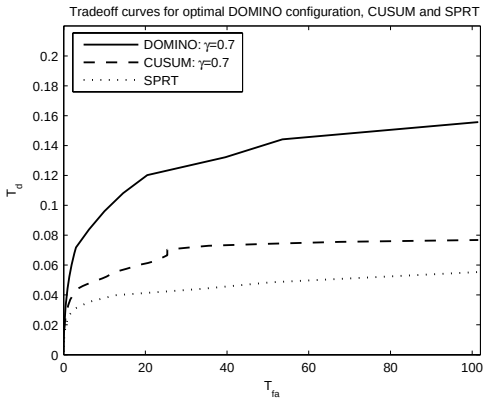


Figure 4.10: Tradeoff curves for best performing DOMINO configuration with $\gamma = 0.7$, best performing CUSUM configuration with $\gamma = 0.7$ and SPRT.

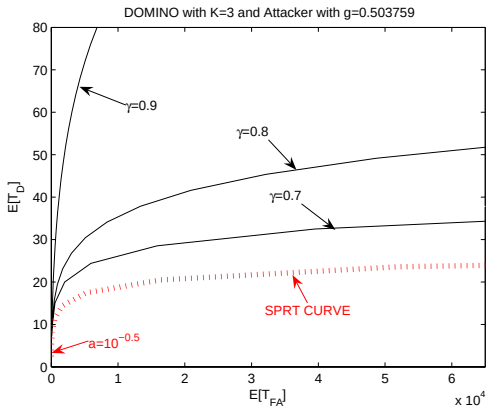


Figure 4.11: Comparison between theoretical and experimental results: theoretical analysis with linear x-axis closely resembles the experimental results.

the SPRT-based detection schemes and provide analytical and intuitive explanation of why the other schemes exhibit suboptimal performance when compared to the SPRT schemes. In addition to that, we offer an extension to the DOMINO algorithm that still preserves its original idea and simplicity, while significantly improving its performance. Our results show the value of doing a rigorous formulation of the problem and providing a formal adversarial model since it can outperform heuristic solutions. We believe our model applies not only to MAC but to a more general adversarial setting. In several practical security applications such as in biometrics, spam filtering, watermarking etc., the attacker has control over the attack distribution and this distribution can be modeled in similar fashion as in our approach.

We now mention some issues for further study. This chapter focused on the performance analysis and comparison of the existing detection schemes. A first issue concerns employment of penalizing functions against misbehaving nodes once an alarm is raised. When an alarm is raised, penalties such as the access point never acknowledging the receipt of the packet (in order to rate-limit the access of the node) or denying access to the medium for a limited period of time should be considered. If constant misbehavior (even after being penalized) is exhibited, the system should apply more severe penalties, such as revocation from the network. A second issue concerns defining alternative objectives of the adversary, such as maximizing throughput while minimizing the probability of detection.

Chapter 5
Secure Data Hiding Algorithms

It will readily be seen that in this case the alleged right of the Duke to the whale was a delegated one from the Sovereign. We must needs inquire then on what principle the Sovereign is originally invested with that right.
-Moby Dick, Herman Melville

I Overview

Digital watermarking allows hidden data, such as a fingerprint or message, to be placed on a media signal (e.g., a sound recording or a digital image). When a watermark detector is given this media signal, it should be able to correctly decode the original embedded fingerprint.

In this chapter we give a final example of the application of our framework to the problem of watermark verification. In the next section we provide a general formulation of the problem. Then in section III we present a watermark verification problem where the adversary is parameterized by a Gaussian distribution. Finally, in section IV we model a watermark verification problem where the adversary is given complete control over its attack distribution.

II General Model

A. Problem Description

The watermark verification problem consists on two main algorithms, an embedding algorithm $\mathcal{E}$ and a detection algorithm $\mathcal{D}$.

- The embedder $\mathcal{E}$ receives as inputs a signal s and a bit m. The objective of the embedder is to produce a signal x with no perceptual loss of information or major differences from s, but that carries also the information about m. The general formulation of this required property is to force the embedder to satisfy the following constraint: $D(x,s) < D_w$, where $D()$ is a distortion function and D_w is an upper bound on the amount of distortion allowed for embedding.

- The detection algorithm $\mathcal{D}$ receives a signal y, which is assumed to be x or an altered version of x either by natural errors (e.g., channel losses) or by an adversary. The detector has to determine if y was embedded with $m = 1$ or not.

- In order to facilitate the detection process, the embedder and the detector usually share a random secret key k (e.g., the seed for a pseudorandom number generator that is used to create an embedding pattern p). This key can be initialized in the devices or exchanged via a secure out of band channel.

B. Adversary Model

- Information available to the adversary: We assume the adversary knows the embedding and detection algorithms $\mathcal{E}$ and $\mathcal{D}$. Furthermore we assume the adversary knows the distribution of K, S, M, however it does not know the particular realizations of these values. That is, the adversary does not know the particular realization of k, s, or m and therefore does not know the input arguments of $\mathcal{E}$.
- Capabilities of the adversary: The adversary is a man-in-the-middle between the embedder and the detector. Therefore it can intercept the signal x produced by $\mathcal{E}$, and can send in its place, a signal y to the detector. However the output of the adversary should satisfy a distortion constraint $D(y,s) < D_a$, since y should be perceptually similar to s.

C. Design Space

The previous description can be summarized by the following diagram:

$$\begin{array}{lcll} \underline{\mathcal{E}_k(m,s)} & & \underline{\mathcal{A}} & \underline{\mathcal{D}_k} \\ x \leftarrow f(x|s,m,k) & \xrightarrow{\quad x \quad} & & \\ & & y \leftarrow f(y|x) & \xrightarrow{\quad y \quad} \\ & & & \ln \frac{f(y|k,m=1)}{f(y|k,m=0)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau \end{array}$$

Where $i \leftarrow f(i|j)$ refers to i being sampled from a distribution with pdf $f(i|j)$. Alternatively i can be understood to be the output of an efficient *probabilistic* algorithm with input j and output i. Therefore from now on, we will use $\mathcal{E}_k(m,s)$ and $\mathcal{A}(y)$ interchangeably with $f(x|s,m,k)$ and $f(y|x)$ (respectively).

Notice that an *optimal* detection algorithm (optimal in the sense that it minimizes several evaluation metrics such as the probability of error, the expected cost or the probability of missed positives given an upper bound on the probability of false positives) is the log-likelihood ratio test $\ln \frac{f(y|k,m=1)}{f(y|k,m=0)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau$, where H_i denotes the hypothesis that $m = i$, and τ depends on the evaluation metric. If the log-likelihood ratio is greater than τ then the detector decides for $m = 1$ and if it less than τ, the detector decides for $m = 0$ (if it is equal to τ the detector randomly selects the hypothesis based on the evaluation metric being considered).

Of course, in order to implement this optimal detection strategy, the detector requires the knowledge of $f(y|k,m)$, which can be computed as follows:

$$\begin{aligned} f(y|k,m) &= \int_S \int_X f(y|s,x,k,m) f(x|s,k,m) f(s) dx ds \\ &= \int_S \int_X f(y|x) f(x|s,k,m) f(s) dx ds \end{aligned}$$

Therefore an optimal detection algorithm requires knowledge of the embedding distribution $f(x|s,k,m)$, the attack algorithm $f(y|x)$, and of the pdf of s: $f(s)$. Note that $f(s)$ is fixed, since neither the embedding algorithm or the adversary can control it. As a result, the overall performance of the detection scheme is only a function of the variable $f(y|x)$ (i.e., on the adversary $\mathcal{A}$) and on the variable $f(x|s,k,m)$ (i.e., the embedding algorithm $\mathcal{E}$). The performance can then be evaluated by a metric which takes into account the two variable parameters: $\Psi(\mathcal{E},\mathcal{A})$.

Assuming $\Psi(\mathcal{E},\mathcal{A})$ represent the losses of the system, our task is to find $\mathcal{E}^*$ to minimize Ψ. However since $\mathcal{A}$ is unknown and arbitrary, the natural choice is to find the embedding scheme $\mathcal{E}^*$ that minimizes the possible damage that can be done by an adversary $\mathcal{A}^*$:

$$(\mathcal{E}^*,\mathcal{A}^*)\arg\min_{\mathcal{E}\in\mathcal{F}_{D_w}}\max_{\mathcal{A}\in\mathcal{F}_{D_a}}\Psi(\mathcal{E},\mathcal{A}) \tag{5.1}$$

where the *feasible design space* $\mathcal{F}_{D_w}$ and the *feasible adversary class* $\mathcal{F}_{D_a}$ consist on embedding distributions and least favorable attack distributions that satisfy certain distortion constraints.

This formulation guarantees that

$$\forall\mathcal{A}\quad\Psi(\mathcal{E}^*,\mathcal{A})\leq\Psi(\mathcal{E}^*,\mathcal{A}^*)$$

whenever possible we are also interested in finding out if $\mathcal{E}^*$ is indeed the best embedding distribution against $\mathcal{A}^*$, and therefore we would like to satisfy a saddle point equilibrium:

$$\Psi(\mathcal{E}^*,\mathcal{A})\leq\Psi(\mathcal{E}^*,\mathcal{A}^*)\leq\Psi(\mathcal{E},\mathcal{A}^*)$$

D. Evaluation metric

The evaluation metric should reflect good properties of the system for the objective it was designed. For our particular case we are going to be interested in the probability of error $\Pr[\mathcal{D}_k(y)\neq m]$ when the following random experiment is performed: k is sampled from a uniform distribution in $\{0,1\}^n$ (where n is the length of k), s is assumed to be sampled from a distribution with pdf $f(s)$, m is sampled from its prior distribution $f(m)$, and then the embedder algorithm and the adversary algorithm are executed. This random experiment is usually expressed in the following notation:

$$\Psi(\mathcal{E},\mathcal{A})\equiv\Pr[k\leftarrow\{0,1\}^n;s\leftarrow f(s);m\leftarrow f(m);x\leftarrow\mathcal{E}_k(s,m);y\leftarrow\mathcal{A}(x):\mathcal{D}_k(y)\neq m] \tag{5.2}$$

The min-max formulation with this evaluation metric minimizes the damage of an adversary whose *objective* is to produce a signal y that removes the embedded information m. In particular the adversary wants the detection algorithm on input y to make an incorrect decision on whether the original signal s was embedded with $m=1$ or not.

III Additive Watermarking and Gaussian Attacks

The model described in the previous section is often intractable (based on the specific objective function, distortion functions, prior distributions etc.) and therefore several approximations are made in order to obtain a solution.

For example, in practice there exists several popular embedding algorithms such as spread spectrum watermarking and QIM watermarking, and researchers often try to optimize the parameters of these embedding schemes, as opposed to finding new embedding algorithms. Furthermore modeling an adversary that can select any algorithm to produce its output is also very difficult because we have to consider non-parametric and arbitrary non-linear attacks. Therefore researchers often assume a linear (additive) adversary that is parameterized by a Gaussian random process. This assumption is motivated by several arguments, including information theoretic arguments claiming a Gaussian distribution is the least favorable noise in a channel, or as an approximation given the central limit theorem.

In this chapter we follow one such model originally proposed by [46]. Contrary however to the results in [46], we relax two assumptions. First, we relax the assumption of spread spectrum watermarking and instead search for the optimal embedding algorithm in this model formulation. Secondly, we relax the assumption of the diagonal processors (an assumption mostly due to the fact that the embedding algorithm used spread spectrum watermarking) and obtain results for the general case. The end result is that our algorithms achieve a lower objective function value than [46] for any possible attacker in the feasible attacker class.

In the following section we describe the model of the problem and obtain our results. Then in section E we discuss our results and compare them to [46].

A. *Mathematical Model*

Given $s \in \mathbb{R}^N$ and $m \in \{0,1\}$, we assume an additive embedder $\mathcal{E}$ that outputs $x = \Phi(s + pm)$, Φ is an $N \times N$ matrix and where $p \in \mathbb{R}^N$ is a pattern sampled from a distribution with pdf $h(p)$. Since p is the only random element in the watermarking algorithm, it is assumed to be dependent on the key k, and therefore from now on we will replace k with p without loss of generality.

The attacker $\mathcal{A}$ is modeled by $y = \Gamma x + e$, where Γ is an $N \times N$ matrix and e is a zero-mean (since any non-zero mean attack is suboptimal [46]) Gaussian random vector with correlation matrix R_e.

Finally, the detection algorithm has to perform the following hypothesis test:

$$\begin{aligned} H_0 : y &= \Gamma\Phi s + e \\ H_1 : y &= \Gamma\Phi s + e + \Gamma\Phi p \end{aligned}$$

If the objective function $\Psi(\mathcal{E}, \mathcal{A})$ the detector wants to minimize is the probability of error, then we know that an optimal detection algorithm is the log-likelihood ratio test. By assuming that the two hypothesis are equally likely, we find that $\tau = 0$. Furthermore in order to compute $f(y|p,m)$ it is assumed s is a Gaussian random vector with zero mean (zero mean is assumed without loss of generality) and correlation matrix R_s.

The following diagram summarizes the model

$$\begin{array}{lll} \underline{\mathcal{E}_p(m,s)} & \underline{\mathcal{A}} & \underline{\mathcal{D}_p} \\ x = (s + pm)\Phi \xrightarrow{\quad x \quad} & & \\ & y = \Gamma x + e \xrightarrow{\quad y \quad} & \\ & & p^t\Upsilon^t R_y^{-1} y - \frac{1}{2} p^t \Upsilon^t R_y^{-1} \Upsilon p \underset{H_0}{\overset{H_1}{\gtrless}} 0 \end{array}$$

Where $R_y = \Gamma\Phi R_s \Phi^t \Gamma^t + R_e$, and $\Upsilon = \Gamma\Phi$.

The distortion constraints that $\mathcal{E}$ and $\mathcal{A}$ need to satisfy are selected to be the squared error distortion:

$$\mathcal{F}_{D_w} = \left\{ \mathcal{E} : \mathbb{E}||X - S||^2 \leq N D_w \right\}$$

and

$$\mathcal{F}_{D_a} = \left\{ \mathcal{A} : \mathbb{E}||Y - S||^2 \leq N D_a \right\}$$

where $\mathcal{E} = (\Phi, R_p, h)$ and $\mathcal{A} = (\Gamma, R_e)$.

B. Optimal Embedding Distribution

In the model described in the previous section, the probability of error can be found to be:

$$\Psi(\mathcal{E},\mathcal{A}) = \Pr[\mathcal{D}_p(y) \neq m] = \mathbb{E}_p\left[Q\left(\sqrt{p^t \Omega p}\right)\right] = \int Q\left(\sqrt{p^t \Omega p}\right) h(p) dp$$

where

$$\Omega = \frac{1}{2}\Phi^t \Gamma^t (\Gamma \Phi R_s \Phi^t \Gamma^t + R_e)^{-1} \Gamma \Phi$$

and where p is the random pattern (watermark) and $h(p)$ is its unknown pdf. Ω is a function of the signal and noise covariances R_s, R_e, the watermark covariance R_p and the scaling matrices Γ and Φ. If we *fix* all these quantities then *we would like to determine the form of* $h(p)$ *that will minimize the error probability* (since this is the goal of the decision maker).

To solve the previous problem we rely on a the following property of the $Q(\cdot)$ function, that can be verified by direct differentiation

Lemma 3: The function $Q(\sqrt{x})$ *is a convex function of* x.

Now we can use this convexity property and apply Jensen's inequality and conclude that

$$\mathbb{E}_x[Q(\sqrt{x})] \geq Q\left(\sqrt{\mathbb{E}_x[x]}\right);$$

we have equality iff x is a constant with probability 1 (wp1). Using this result in our case we get

$$\mathbb{P}_e = \mathbb{E}_p\left[Q\left(\sqrt{p^t \Omega p}\right)\right] \geq Q\left(\sqrt{\mathbb{E}_p\left[p^t \Omega p\right]}\right) = Q\left(\sqrt{\mathrm{tr}\{\Omega R_p\}}\right). \tag{5.3}$$

Relation (5.3) provides a *lower bound* on the error probability for *any* pdf (which of course satisfies the covariance constraint). We have equality in (5.3) iff wp1 we have that

$$p^t \Omega p = \mathrm{tr}\{\Omega R_p\}. \tag{5.4}$$

In other words *every realization* of p (that is every watermark p) must satisfy this equality. Notice that if we can find a pdf for p which can satisfy (5.4) *under the constraint* that $\mathbb{E}[pp^t] = R_p$ (remember we fixed the covariance R_p), then we will attain the lower bound in Equation (5.3).

To find a random vector p that achieves what we want, we must do the following. Consider the SVD of the matrix

$$R_p^{1/2} \Omega R_p^{1/2} = U \Sigma U^t \tag{5.5}$$

where U orthonormal and $\Sigma = \mathrm{diag}\{\sigma_1, \ldots, \sigma_K\}$, diagonal with nonnegative elements. The nonnegativity of σ_i is assured because the matrix is nonnegative definite. Let A be a random vector with i.i.d. elements that take the values ± 1 with probability 0.5. For every vector A we can then define an embedding vector p as follows

$$p = R_p^{1/2} U A \tag{5.6}$$

Let us see if this definition satisfies our requirements. First consider the covariance matrix which must be equal to R_p. Indeed we have

$$\mathbb{E}[pp^t] = R_p^{1/2} U \mathbb{E}[AA^t] U^t R_p^{1/2} = R_p^{1/2} U \boldsymbol{I} U^t R_p^{1/2} = R_p^{1/2} \boldsymbol{I} R_p^{1/2} = R_p,$$

where we used the independence of the elements of the vector A and the orthonormality of U ($\boldsymbol{I}$ denotes the identity matrix). So our random vector has the correct covariance structure. Let us now see whether it also satisfies the constraint that $p^t \Omega p = \mathrm{tr}\{\Omega R_p\}$ wp1. Indeed for every realization of the random vector A we have

$$\begin{aligned} p^t \Omega p &= A^t U^t R_p^{1/2} \Omega R_p^{1/2} U A = A^t \Sigma A = A_1^2 \sigma_1 + A_2^2 \sigma_2 + \cdots + A_K^2 \sigma_K \\ &= \sigma_1 + \sigma_2 + \cdots + \sigma_K, \end{aligned}$$

where we use the fact that the elements A_i of A are equal to ± 1. Notice also that

$$\mathrm{tr}\{\Omega R_p\} = \mathrm{tr}\{R_p^{1/2} \Omega R_p^{1/2}\} = \mathrm{tr}\{U \Sigma U^t\} = \mathrm{tr}\{\Sigma U^t U\} = \mathrm{tr}\{\Sigma\} = \sigma_1 + \cdots + \sigma_K,$$

which proves the desired equality. So we conclude that, although p is a random vector (our possible watermarks), *all* its realizations satisfy the equality

$$p^t \Omega p = \mathrm{tr}\{\Omega R_p\}.$$

This of course suggests that this specific choice of watermarking attains the lower bound in (5.3).

B.1 Summary

We have found the *optimum* embedding distribution. It is a random mixture of the columns of the matrix $R_p^{1/2} U$ of the form $R_p^{1/2} U A$, where A is a vector with elements ± 1.

This of course suggests that we can have 2^N different patterns. R_p is the *final* matrix we end up from the max-min game and U is the SVD of the corresponding *final* matrix $R_p^{1/2} \Omega R_p^{1/2}$.

Once we are given h^*, the game the embedder and the attacker play is the following:

$$\max_{R_p, \Phi} \min_{R_e, \Gamma} \mathrm{tr}\{\Omega R_p\}$$

More specifically:

$$\max_{R_p, \Phi} \min_{R_e, \Gamma} \mathrm{tr}\{(\Gamma \Phi R_s \Phi^t \Gamma^t + R_e)^{-1} \Gamma \Phi R_p \Phi^t \Gamma^t\} \tag{5.7}$$

Subject to the distortion constraints:

$$\mathrm{tr}\{(\Phi - I) R_s (\Phi - I)^t + \Phi R_p \Phi^t\} \leq N D_w \tag{5.8}$$

$$\mathrm{tr}\{(\Gamma \Phi - I) R_s (\Gamma \Phi - I)^t + \Gamma \Phi R_p \Phi^t \Gamma^t + R_e\} \leq N D_a \tag{5.9}$$

C. Least Favorable Attacker Parameters

C.1 Minimization with respect to R_e

Assuming $\Upsilon = \Gamma\Phi$ is fixed, we start by minimizing (**??**) with respect to R_e. This minimization problem is addressed with the use of variational techniques. Let $R_e^\varepsilon = R_e^o + \varepsilon\Delta$. By forming the Lagrangian of the optimization problem (**??**) under constraint (**??**), the goal of the attacker is to minimize with respect to ε the following objective function:

$$f(\varepsilon) = \mathrm{tr}\{(\Upsilon R_s \Upsilon^t + R_e^\varepsilon)^{-1} \Upsilon R_p \Upsilon^t\} + \mu(\mathrm{tr}\{(\Upsilon - I)R_s(\Upsilon - I)^t + \Upsilon R_p \Upsilon^t + R_e^\varepsilon\} - ND_a) \quad (5.10)$$

A necessary condition for the optimality of R_e^* is when

$$\frac{df(\varepsilon)}{d\varepsilon}|_{\varepsilon=0} = 0$$

which implies

$$-\mathrm{tr}\{(\Upsilon R_s \Upsilon^t + R_e^*)^{-1}\Delta(\Upsilon R_s \Upsilon^t + R_e^*)^{-1}\Upsilon R_p \Upsilon^t\} + \mu\mathrm{tr}\{\Delta\} = 0$$

which must be zero for any Δ, and thus we need that

$$(\Upsilon R_s \Upsilon^t + R_e^*)^{-1}\Upsilon R_p \Upsilon^t (\Upsilon R_s \Upsilon^t + R_e^*)^{-1} = \mu I$$

from where we solve for $R_e^* = \frac{1}{\sqrt{\mu}}(\Upsilon R_p \Upsilon^t)^{1/2} - \Upsilon R_s \Upsilon$

The dual problem is then to maximize with respect to μ the following function:

$$\sqrt{\mu}\mathrm{tr}\{(\Upsilon R_p \Upsilon^t)^{1/2}\} + \mu(\mathrm{tr}\{(\Upsilon - I)R_s(\Upsilon - I)^t + \Upsilon R_p \Upsilon^t + (\mu)^{-1/2}(\Upsilon R_p \Upsilon^t)^{1/2} - \Upsilon R_s \Upsilon^t\} - ND_a)$$

which after some simplification becomes

$$2\sqrt{\mu}\mathrm{tr}\{(\Upsilon R_p \Upsilon^t)^{1/2}\} - 2\mu\mathrm{tr}\{\Upsilon R_s\} + \mu\mathrm{tr}\{Rs\} + \mu\mathrm{tr}\{\Upsilon R_p \Upsilon^t\} - \mu ND_a$$

taking the derivative with respect to μ and equating to zero we obtain:

$$\frac{1}{\sqrt{\mu}} = \frac{2\mathrm{tr}\{\Upsilon R_s\} - \mathrm{tr}\{R_s\} - \mathrm{tr}\{\Upsilon R_p \Upsilon^t\} + ND_a}{\mathrm{tr}\{(\Upsilon R_p \Upsilon^t)^{1/2}\}}$$

C.2 Minimization with respect to Γ

Since $\Upsilon = \Gamma\Phi$ we note that the attacker can completely control Υ by an appropriate choice of Γ (assuming Φ is invertible), therefore we only need to consider the minimization over Υ of the following function:

$$\frac{(\mathrm{tr}\{(\Upsilon R_p \Upsilon^t)^{1/2}\})^2}{2\mathrm{tr}\{\Upsilon R_s\} - \mathrm{tr}\{R_s\} - \mathrm{tr}\{\Upsilon R_p \Upsilon^t\} + ND_a}$$

Proceeding similarly to the previous case, we use variational techniques with $\Upsilon^\varepsilon = \Upsilon_o + \varepsilon\Delta$. After deriving the objective function with respect to ε and equating to zero, we obtain the following equation:

$$\mathrm{tr}\{(\Delta R_p {\Upsilon_o}^t + \Upsilon_o R_p \Delta^t)(\Upsilon_o R_p {\Upsilon_o}^t)^{-1/2}\}C_d - (2\mathrm{tr}\{\Delta R_s\} - \mathrm{tr}\{\Delta R_p {\Upsilon_o}^t + \Upsilon_o R_p \Delta^t\})C_n = 0$$

where C_d and C_n are scalar factors not dependent of Δ (and will be determined later.)
Since the above equation must be equal to zero for any Δ we need that

$$2C_d(R_p\Upsilon_o^t(\Upsilon_o R_p \Upsilon^t)^{-1/2}) - 2C_n(R_s - R_p\Upsilon_o^t) = 0$$

or alternatively, if we let $C = C_d/C_n$

$$c(\Upsilon_o R_p \Upsilon_o^t)^{1/2} + \Upsilon_o R_p \Upsilon_o^t = \Upsilon R_s$$

Letting $\Sigma = \Upsilon_o Rp^{1/2}$ and $A = R_p^{-1/2} R_s$ we obtain

$$C(\Sigma\Sigma^t)^{1/2} = \Sigma A - \Sigma\Sigma^t$$

after squaring both sides we have

$$C^2\Sigma^t = (A - \Sigma^t)\Sigma(A - \Sigma^t)$$

or equivalently,

$$(A - \Sigma^t)\Sigma(A - \Sigma^t) - C^2\Sigma^t = 0$$

D. Optimal Embedding Parameters

$$\max_{\Phi, R_p} \frac{\left(\text{tr}\left\{(\Phi R_p \Phi^t)^{1/2}\right\}\right)^2}{2\text{tr}\{\Phi R_s\} - \text{tr}\{R_s\} - \text{tr}\left\{\Phi R_p \Phi^t\right\} + ND_a} \tag{5.11}$$

Subject to:

$$\text{tr}\left\{(\Phi - I)R_s(\Phi - I)^t + \Phi R_p \Phi^t\right\} \leq ND_w \tag{5.12}$$

For any Φ and any R_p we have by Schwarz inequality that:

$$\frac{\left(\text{tr}\left\{(\Phi R_p \Phi^t)^{1/2}\right\}\right)^2}{2\text{tr}\{\Phi R_s\} - \text{tr}\{R_s\} - \text{tr}\left\{\Phi R_p \Phi^t\right\} + ND_a} \tag{5.13}$$

$$\leq \quad \frac{N\text{tr}\{\Phi R_s \Phi^t\}}{2\text{tr}\{\Phi R_s\} - \text{tr}\{R_s\} - \text{tr}\left\{\Phi R_p \Phi^t\right\} + ND_a} \tag{5.14}$$

$$\tag{5.15}$$

Equality is achieved if and only if $\Phi R_p \Phi^t = \kappa I$ (i.e., $R_p^* = \kappa(\Phi^t\Phi)^{-1}$), where κ is a constant that can be determined by the following arguments. Notice first that Equation 5.14 is an increasing function of $\text{tr}\{\Phi R_p \Phi^t\}$. The maximum value is therefore achieved when Equation is satisfied with equality. Solving for κ from this constraint we obtain

$$\kappa = \frac{ND_w - \text{tr}\{(\Phi - I)R_s(\Phi - I)^t\}}{N}$$

Replacing the values of κ and R_p into the original objective function and letting λ be the maximum value the objective function can achieve (as a function of Φ), we have:

$$\frac{N(ND_w - \mathrm{tr}\{(\Phi - I)R_s(\Phi - I)^t\})}{2\mathrm{tr}\{\Phi R_s\} - \mathrm{tr}\{R_s\} + ND_a - ND_w + \mathrm{tr}\{(\Phi - I)R_s(\Phi - I)^t\}} \leq \lambda$$

After some algebraic manipulations we obtain the following:

$$\frac{ND_w}{\lambda+1} - \frac{\lambda(D_a - D_w)}{\lambda+1} \leq \mathrm{tr}\left\{(\Phi - (\lambda+1)^{-1}I)R_s(\Phi - (\lambda+1)^{-1}I)^t\right\} + \frac{\lambda \mathrm{tr}\{R_s\}}{(\lambda+1)^2}$$

We can see that the minimum value the right hand side of this equation can achieve (as a function of Φ) is when $\Phi^* = (\lambda+1)^{-1}I$. With Φ^*, the following equation can be used to solve for λ:

$$\frac{\lambda}{\lambda+1}\mathrm{tr}\{R_s\} + \lambda(D_a - D_w) = ND_w$$

Solving for λ we obtain

$$\lambda = \frac{D_w - D_a + ND_w - \mathrm{tr}\{R_s\} + \sqrt{4(D_a - D_w)ND_w + (D_a - D_w - ND_w + \mathrm{tr}\{R_s\})^2}}{2(D_a - D_w)}$$

E. Discussion

Notice that so far we have solved the problem in the following way:

$$\min_{\Phi, R_p} \max_{\Gamma, R_e} \min_{h} \Psi(\mathcal{E}, \mathcal{A}) \tag{5.16}$$

where (as we mentioned before) $\mathcal{E} = (h, R_p, \Phi)$ and $\mathcal{A} = (\Gamma, R_e)$. This means that given Φ and R_p, the adversary will select Γ and R_e in order to maximize the probability of error, and then given the parameters chosen by the adversary we finally find the embedding distribution h minimizing the probability of error.

The problem with this solution is that in practice, the embedding algorithm will be given in advance and the adversary will have the opportunity of changing its behavior based on the given embedding algorithm (including h).

For notational simplicity assume Φ and R_p are fixed, so we can replace $\mathcal{E}$ with h in the remaining of this chapter. Furthermore let $h(\mathcal{A})$ denote the embedding distribution as a function of the parameters $\mathcal{A} = (\Gamma, R_e)$ (recall that h depends on $\mathcal{A}$ by the selection of U in Equation 5.6). Now we can express easily the problem we have solved:

$$\forall \mathcal{A} \;\; \Psi(h^*(\mathcal{A}), \mathcal{A}) \leq \Psi(h(\mathcal{A}), \mathcal{A})$$

This is true in particular for $\mathcal{A}^*$, the solution to the full optimization problem from Equation 5.16. Moreover, the above is also true for the distribution used in the previous work [46], which assumed a Gaussian embedding distribution h^G:

$$\forall \mathcal{A} \;\; \Psi(h^*(\mathcal{A}), \mathcal{A}) \leq \Psi(h^G, \mathcal{A})$$

Notice also that in [46], the solution obtained was

$$\mathcal{A}^G = \arg \max_{\mathcal{A}} \Psi(h^G, \mathcal{A}) \tag{5.17}$$

Due to some approximations done in [46], $\Psi(h^G, \mathcal{A})$ turns out to be the same objective function given in Equation 5.7. Furthermore in [46] there were further approximations in order to obtain linear processors (diagonal matrices). In this work we relaxed this assumption in order to obtain the full solution to Equation 5.7. Therefore the general solution (without extra assumptions such as diagonal matrices) in both cases is the same:

$$A^G = \arg \max_{\mathcal{A}} \Psi(h^G, \mathcal{A}) = A^* = \arg \max_{\mathcal{A}} \left\{ \min_{h} \Psi(h, \mathcal{A}) \right\} \tag{5.18}$$

One of the problems with our solution however is that there might exist $\mathcal{A}'$ such that

$$\Psi(h^*(\mathcal{A}*), \mathcal{A}^*) < \Psi(h^*(\mathcal{A}'), \mathcal{A}')$$

However, even in this case it is easy to show that h^* is still better than h^G, since the performance achieved by h^G is not as good as the performance obtained with h^*, even for any other $\mathcal{A}$:

$$\max_{\mathcal{A}} \Psi(h^*(\mathcal{A}), \mathcal{A}) < \max_{\mathcal{A}} \Psi(h^G, \mathcal{A})$$

The main problem is that in order to obtain this optimal performance guarantee, the embedding distribution h^* needs to know the adversary final strategy of the adversary $\mathcal{A}$. In particular we are interested in two questions. With regards to the previous work in [46] we would like to know if the following is true:

$$\forall \mathcal{A}\, \Psi(h^*(\mathcal{A}^*), \mathcal{A}) \leq \Psi(h^G, \mathcal{A}^*) \tag{5.19}$$

that is, once we have fixed the operating point $\mathcal{A}^*$ (the optimal adversary according to Equation 5.7) there is no other adversarial strategy that will make h^* perform worse than the previous work.

The second question is in fact more general and it relates to the original intention of minimizing the worst possible error created by the adversary:

$$\min_{h} \max_{\mathcal{A}} \Psi(h, \mathcal{A}) \tag{5.20}$$

yet we have only solved the problem in a way where h is dependent on $\mathcal{A}$:

$$(h^*, \mathcal{A}^*) = \arg \max_{\mathcal{A}} \min_{h} \Psi(h, \mathcal{A})$$

A way to show that $(h^*, \mathcal{A}^*)$ satisfies Equation 5.20 (and therefore also satisfy Equation 5.19) is to show that the pair $(h^*, \mathcal{A}^*)$ forms a saddle point equilibrium:

$$\forall (h, \mathcal{A})\ \ \Psi(h^*, \mathcal{A}) \leq \Psi(h^*, \mathcal{A}^*) \leq \Psi(h, \mathcal{A}^*) \tag{5.21}$$

To be more specific let $\mathcal{E}$ denote again the triple (h, R_p, Φ). Then we are interested in showing that

$$\forall (\mathcal{E}, \mathcal{A})\ \ \Psi(\mathcal{E}^*, \mathcal{A}) \leq \Psi(\mathcal{E}^*, \mathcal{A}^*) \leq \Psi(\mathcal{E}, \mathcal{A}^*) \tag{5.22}$$

where $(\mathcal{E}^*,\mathcal{A}^*) = (h^*, R_p^*, \Phi^*, R_e^*, \Gamma^*)$ is the solution to Equation (5.16).

It is easy to show how the right hand side inequality of Equation (5.22) is satisfied:

$$\begin{aligned}
\Psi(\mathcal{E},\mathcal{A}^*) &= \mathbb{E}_p\left[Q\left(\sqrt{p^t\frac{1}{2}\Phi^t\Gamma^{*t}(\Gamma^*\Phi R_s\Phi^t\Gamma^{*t}+R_e^*)^{-1}\Gamma^*\Phi p}\right)\right] \\
&\geq Q\left(\sqrt{R_p\frac{1}{2}\Phi^t\Gamma^{*t}(\Gamma^*\Phi R_s\Phi^t\Gamma^{*t}+R_e^*)^{-1}\Gamma^*\Phi}\right) \quad \text{by Jensen's inequality} \\
&\geq Q\left(\sqrt{R_p^*\frac{1}{2}\Phi^{*t}\Gamma^{*t}(\Gamma^*\Phi^* R_s\Phi^{*t}\Gamma^{*t}+R_e^*)^{-1}\Gamma^*\Phi^*}\right) \quad \text{by Equation (5.7)} \\
&= \Psi(\mathcal{E}^*,\mathcal{A}^*) \quad \text{by the definition of } h^*
\end{aligned}$$

The left hand side of Equation (5.22) is more difficult to satisfy. A particular case where it is satisfied is the *scalar* case, i.e., when $N=1$. In this case we have the following:

$$\begin{aligned}
\Psi(\mathcal{E}^*,\mathcal{A}^*) &= Q\left(\sqrt{\frac{R_p^*(\Phi^*\Gamma^*)^2}{2((\Gamma^*\Phi^*)^2R_s+R_e^*)}}\right) \\
&\geq Q\left(\sqrt{\frac{R_p^*(\Phi^*\Gamma)^2}{2((\Gamma\Phi^*)^2R_s+R_e)}}\right) \quad \text{by Equation (5.7)} \\
&= \mathbb{E}_p\left[Q\left(\sqrt{\frac{p^2(\Phi^*\Gamma)^2}{2((\Gamma\Phi^*)^2R_s+R_e)}}\right)\right] \quad \text{Since } p \text{ is independent of } \mathcal{A} \\
&= \Psi(\mathcal{E}^*,\mathcal{A}^*)
\end{aligned}$$

The independence of p in the scalar case comes from the fact that Equation (5.6) yields in this case $p=\sqrt{R_p}$ with probability $\frac{1}{2}$ and $p=-\sqrt{R_p}$ with probability $\frac{1}{2}$. With this distribution Equation (5.4) is always satisfied (since the adversary has no control over it).

This result can in fact be seen as a counterexample against the optimality of spread spectrum watermarking against Gaussian attacks: if the attack is Gaussian, then the embedding distribution should not be a spread spectrum watermarking, or conversely, if the embedding distribution is spread spectrum, then the attack should not be a Gaussian attack.

In future work we plan to investigate under which conditions or assumptions is the left inequality in Equation (5.22) satisfied. An easier goal we also plan to investigate is whether Equation (5.19) is true, since this will also show an improvement over previous work. We also plan to extend the work to other evaluation metrics, such as the case when one of the errors is more important than the other. In this case we can set an arbitrary level of false alarms and find the parameters of the embedder that maximize the probability of detection while the adversary tries to minimize detection.

IV Towards a Universal Adversary Model

In the previous section we attempted to relax the usual assumptions regarding the optimal embedding distributions (Spread Spectrum or QIM) and find new embedding distributions that achieve better performance against attacks. The problem with the formulation in the previous section is that the optimal embedding distribution h^* will depend on the strict assumptions made to keep the problem tractable. In particular it assumes the correctness of the source signal model

$f(s)$ and even more troubling, it assumes the adversary will perform a Gaussian and scaling attack only. This limitation on the capabilities of the adversary is a big problem in practice, since any data hiding algorithm that is shown to perform well under any parametric adversary (e.g., Gaussian attacks) will give a false sense of security, since in reality the adversary will never be confined to create only Gaussian attacks or follow any other parametric distribution prescribed by the analysis.

In this section we are going to focus in another version of the embedding detection problem: non-Blind watermarking. This formulation is very important for several problems such as fingerprinting and traitor tracing. In non-blind watermarking both the embedder and the detector have access to the source signal s, and therefore the problem can again be represented as:

$$\begin{array}{ccccc}
\underline{\mathcal{E}_k(m,s)} & & \underline{\mathcal{A}} & & \underline{\mathcal{D}_k(s)} \\
x \leftarrow f(x|k,m,s) & \xrightarrow{\quad x \quad} & & & \\
 & & y \leftarrow f(y|x) & \xrightarrow{\quad y \quad} & \\
 & & & & \ln \frac{f(y|s,k,m=1)}{f(y|s,k,m=0)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau
\end{array}$$

where $\mathcal{A}$ should satisfy some distortion constraint, for example for quadratic distortion constraints: $\mathcal{A} \in \mathcal{F}_D : \left\{\mathcal{A} : \mathbb{E}[(x-y)^2] \leq D\right\}$

Our main objective is to model and understand the optimal strategy that a non-parametric adversary can do. To the best of our knowledge this is the first attempt to model this all powerful adversary.

In order to gain a better insight into the problem we are going to start with the scalar case: $N = 1$, or in particular s, x and y are in $\mathbb{R}$. Furthermore we assume $\mathcal{E}$ is fixed and parameterized by a distance d between different embeddings: that is, for all k and s $d = |\mathcal{E}_k(1,s) - \mathcal{E}_k(0,s)|$. Since for every output of the attacker $y \leftarrow f(y|x)$ there exists a random realization of a with pdf h such that $y = x + a$, we can replace the adversarial model with an additive random variable a sampled from an attacker distribution h. Finally, the decision function ρ will output an estimate of m: $m = 0$ or $m = 1$ given the output of the adversary: y.

$$\begin{array}{ccccc}
\underline{\mathcal{E}(m,s)} & & \underline{\mathcal{A}} & & \underline{\mathcal{D}(s)} \\
 & \xrightarrow{\quad x \quad} & & & \\
 & & y = x + a & \xrightarrow{\quad y \quad} & \\
 & & & & \rho(y)
\end{array}$$

Having fixed the embedding algorithm this time, our objective is to find a pair (ρ^*, h^*) such that

$$\forall \rho \text{ and } h \in \mathcal{F}_D \quad \Psi(\rho^*, h) \leq \Psi(\rho^*, h^*) \leq \Psi(\rho, h^*) \tag{5.23}$$

where $\Psi(\rho, h)$ is again the probability of error:

$$\Pr[m \leftarrow \{0,1\}; x = \mathcal{E}(m,s); a \leftarrow h(a) \ : \ \rho(x+a) \neq m]$$

and where $\mathcal{F}_D$ simplifies to $\left\{h : \mathbb{E}[a^2] \leq D, \ \int h(a)da = 1 \text{ and } h \geq 0\right\}$.

Notice that this problem is significantly more difficult than the problem of finding the optimal parameters of an adversary, since in this case we need to perform the optimization over infinite dimensional spaces, because h is a continuous function.

A. On the Necessity of Randomized Decisions for Active Distortion Constraints

Let $x_i = \mathcal{E}(i,s)$. Then we know that to satisfy $\Psi(\rho^*,h^*) \leq \Psi(\rho,h^*)$, ρ^* should select the largest between the likelihood of y given x_1: $f(y|x_1)$ and the likelihood of y given x_0: $f(y|x_0)$, and should randomly flip a coin to decide if both likelihoods are equal (this decision is called *Bayes optimal*). Therefore in order to find the saddle point equilibrium we are going to assume a given ρ and maximize for h (subject to the distortion constraints) and then check to see if ρ is indeed *Bayes optimal*.

Before we obtain a saddle point solution we think it is informative to show our attempt to solve the problem with a non-randomized decision function ρ. A typical non-randomized decision function will divide the decision space into two sets: R and its complement R^c. If $y \in R$ then $\rho(y) = 1$, otherwise $\rho(y) = 0$.

Assume without loss of generality that $d = x_0 - x_1 > 0$. The probability of error can be expressed then as:

$$\begin{aligned}
\Psi(\rho,h) &= \Pr[\rho = 1|M=0]\Pr[M=0] + \Pr[\rho = 0|M=1]\Pr[M=1] \\
&= \frac{1}{2}\left(\int_R h(y-x_0)dy + \int_{R^c} h(y-x_1)dy\right) \\
&= \frac{1}{2}\left(\int_R h(a-d)da + \int_{R^c} h(a)da\right) \\
&= \frac{1}{2}\left(\int_{R-d} h(a)da + \int_{R^c} h(a)da\right) \\
&= \frac{1}{2}\int \left(1_{R-d}(a) + 1_{R^c}(a)\right) h(a)da
\end{aligned}$$

where 1_R is the indicator function for the set R (i.e., $1_R(a) = 1$ if $a \in R$ and $1_R(a) = 0$ otherwise) and where $R-d$ is defined as the set $\{a-d \ : \ a \in R\}$.

The objective function is therefore:

$$\min_{R\in\mathbb{R}} \max_{h\in\mathcal{F}_D} \frac{1}{2}\int \left(1_{R-d}(a) + 1_{R^c}(a)\right) h(a)da$$

Subject to:

$$\mathbb{E}[a^2] \leq D$$

The Lagrangian is

$$L(\lambda,h) = \int \left(\frac{1}{2}\left(1_{R-d}(a) + 1_{R^c}(a)\right) - \lambda a^2\right) h(a)da + \lambda D$$

where $\Psi(\rho,h^*) = L^*(\lambda^*) = L(\lambda^*,h^*)$.

By looking at the form of the Lagrangian in Figure 5.1 (for $\lambda > 0$) it is clear that a necessary condition for optimality is that $R-d \bigcap R^c = \emptyset$, since otherwise, the adversary will put all the mass of h in this interval. Under this condition we assume $R-d = [-\inf, \frac{-d}{2}]$.

Now notice that for $D \geq (\frac{d}{2})^2$, $\lambda^* = 0$, and therefore there will always be an h^* such that $\Psi(\rho,h^*) = \frac{1}{2}$. The interpretation for this case is that the distortion constraints are not strict enough, and the adversary can create error rates up to 0.5. It is impossible to find a saddle point solution for this case, since any ρ will not be Bayesian, and if it is Bayes optimal then h^* is

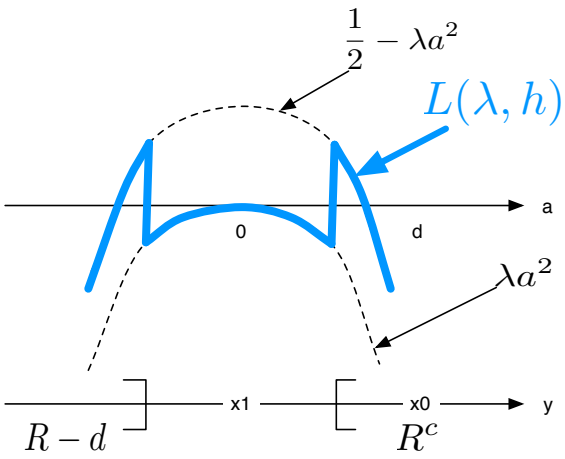


Figure 5.1: Let us define $a = y - x_1$ for the detector. We can now see the Lagrangian function over a, where the adversary tries to distribute the density h such that $L(\lambda, h)$ is maximized while satisfying the constraints (i.e., minimizing $L(\lambda, h)$ over λ.

not a maximizing distribution. However by assuming $R - d = [-\inf, \frac{-d}{2}]$ we guarantee that the probability of error is not greater than 0.5. Having such a high false alarm rate is unfeasible in practice and thus the embedding scheme should be designed with a d such that $D \geq (\frac{d}{2})^2$

Assuming $\lambda > 0$ it is now clear from Figure 5.1 that an optimal solution is for

$$h^*(a) = p_0\delta(-d/2) + p_1\delta(0) + p_2(d/2)$$

where $p_0 + p_1 + p_2 = 1$.

For $D < (\frac{d}{2})^2$, $\lambda^* = \frac{2}{d^2}$ and thus $\Psi(\rho, h^*) = \frac{2D}{d^2}$, where

$$h^*(a) = \frac{2D}{d^2}\delta(-d/2) + \frac{d^2 - 4D}{d^2}\delta(0) + \frac{2D}{d^2}\delta(d/2)$$

Notice however that in this case ρ is not optimal, since the solution assumes that at the boundary between R and R^c, ρ decides for both: $m = 0$ and $m = 1$ at the same time! and thus clearly this is not a Bayes optimal decision rule (in fact this is not a decision at all!). An naive approach to solve this problem is to randomize the decision at the boundary: i.e., to flip an unbiased coin whenever y is in the boundary between R and R^c. This decision will then be Bayes optimal for h^*, however it can be shown that this h^* is not a solution h that maximizes the probability of error and thus we cannot achieve a saddle point solution. In the next section we introduce a more elaborate randomized decision that achieves a saddle point equilibria.

B. Achieving Saddle Point Equilibria with Active Constraints

Let $\rho(a) = 0$ with probability $\rho_0(a)$ and $\rho(a) = 1$ with probability $\rho_1(a)$. In order to have a well defined decision function we require $\rho_0 = 1 - \rho_1$. Consider now the decision function given in Figure 5.2. The Lagrangian is now:

$$L(\lambda, h) = \int \left(\frac{1}{2}(\rho_1(a+d) + \rho_0(a)) - \lambda a^2\right) h(a)da + \lambda D$$

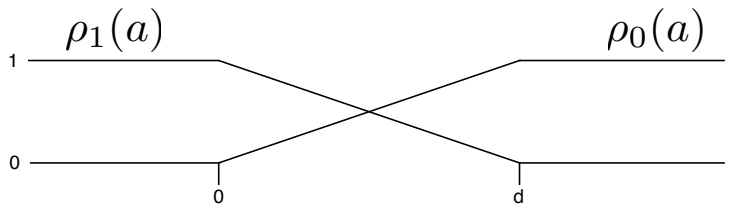


Figure 5.2: Piecewise linear decision function, where $\rho(0) = \rho_0(0) = \rho_1(0) = \frac{1}{2}$

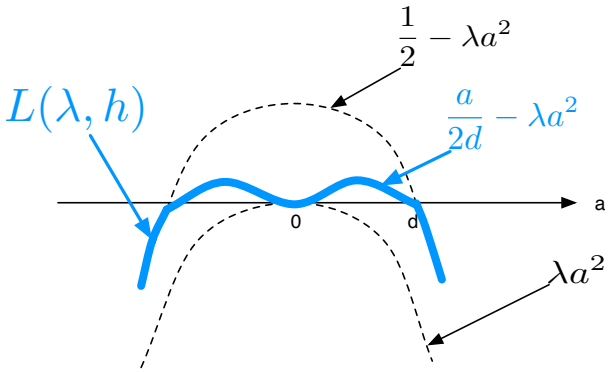


Figure 5.3: The discontinuity problem in the Lagrangian is solved by using piecewise linear *continuous* decision functions. It is now easy to shape the Lagrangian such that the maxima created form a saddle point equilibrium.

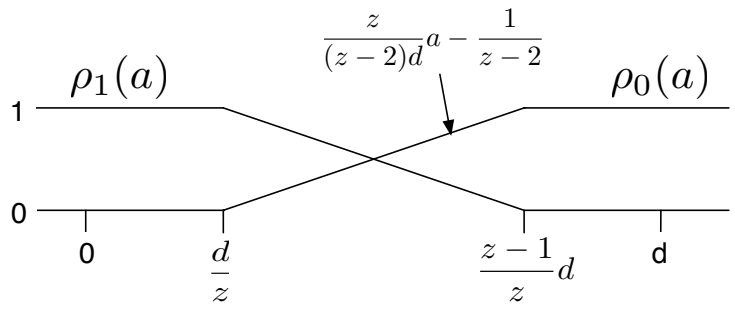


Figure 5.4: New decision function

In order to have active distortion constraints, the maxima of $L(\lambda, h)$ should be in the interval $a \in [-d, d]$. Looking at Figure 5.3 we see that

$$\frac{1}{2}(\rho_1(a+d) + \rho_0(a)) - \lambda a^2$$

achieves its maximum value for $a^* = \pm\frac{1}{4\lambda d}$. Therefore

$$h^*(a) = \frac{1}{2}\left(\delta\left(-\frac{1}{4\lambda d}\right) + \delta\left(\frac{1}{4\lambda d}\right)\right)$$

Notice however that under h^*, ρ will only be Bayes optimal if and only if $\frac{1}{4\lambda d} = \frac{d}{2}$, which occurs if and only if $\lambda^* = \frac{1}{2d^2}$ which occurs if and only if $D = \left(\frac{d}{2}\right)^2$ (since $\lambda^* = \frac{1}{4d\sqrt{D}}$ minimizes the Lagrangian.)

As a summary, for $D = \left(\frac{d}{2}\right)^2$, (ρ^*, h^*) form a saddle point equilibrium when ρ^* is defined as in Figure 5.3 and

$$h^*(a) = \frac{1}{2}\left(\delta\left(-\frac{-d}{2}\right) + \delta\left(\frac{d}{2}\right)\right)$$

Furthermore the probability of error is $\Psi(\rho^*, h^*) = \frac{1}{16\lambda d^2} + \lambda D = \frac{1}{8} + \frac{D}{2d^2} = \frac{1}{4}$.

It is still an open question whether there are saddle point equilibria for the distortion constraint $\mathbb{E}[a^2] < D$ where $D > \frac{d^2}{4}$, however for $D < \frac{d^2}{4}$ we can obtain a saddle point by considering the decision function shown in Figure 5.4. For $z \in (3, \infty)$, the local maxima of the Lagrangian occur for $a = 0$, and $a = \pm\frac{z}{4d(z-2)\lambda}$, where the second value was obtained as the solution to

$$\frac{d}{da}\left[\frac{1}{2}\left(\frac{z}{(z-2)d}a - \frac{1}{z-2}\right) - \lambda a^2\right]_{a^*} = 0$$

i.e., the derivative evaluated at a^* must be equal to zero. By symmetry of the Lagrangian we have that another local maximum occurs at $a = -a^*$.

The value of λ^* that makes all these local maxima the same (and thus gives the opportunity of an optimal attack with three delta functions, one on each local maximum) is the solution to:

$$\frac{1}{2}\left(\frac{z}{(z-2)d}\frac{z}{4d(z-2)\lambda^*} - \frac{1}{z-2}\right) - \lambda^*\left(\frac{z}{4d(z-2)\lambda^*}\right)^2 = \frac{z^2 - 8d^2(x-2)\lambda^*}{16d^2(z-2)^2\lambda^*} = 0$$

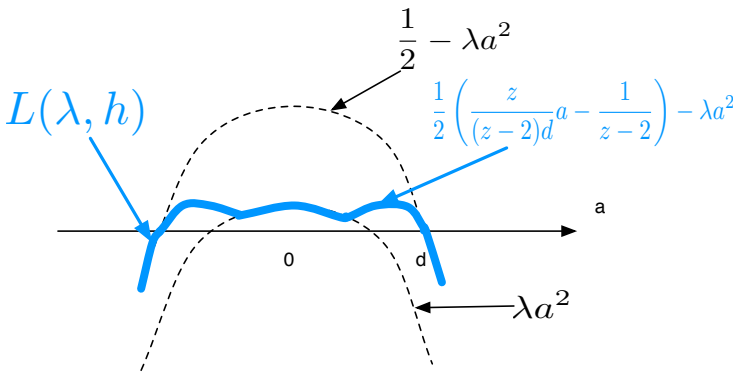


Figure 5.5: With ρ defined in Figure 5.4 the Lagrangian is able to exhibit three local maxima, one of them at the point $a = 0$, which implies that the adversary will use this point whenever the distortion constraints are too severe

which is $\lambda^* = \frac{z^2}{8d^2(z-2)}$. Any other λ would have implied inactive constraints (D too large) or $D = 0$. See Figure 5.5.

The optimal adversary has the form

$$h^*(a) = p_0\delta\left(-\frac{2d}{z}\right) + (1-2p_0)\delta(0) + p_0\delta\left(\frac{2d}{z}\right)$$

We can see that ρ defined as in Figure 5.4, and h^* can only form a saddle point if $z = 4$. Notice also that h^* is an optimal strategy for the adversary as long as

$$\mathbb{E}[a^2] = 2p_0\left(\frac{d}{2}\right)^2 = D$$

That is $p_0 = \frac{2D}{d^2}$. Since the maximum value that p_0 should attain is $\frac{1}{2}$, this implies that this is the optimal strategy for the adversary for any $D \leq \frac{d^2}{4}$. The probability of error for this saddle point equilibrium is

$$\Psi(\rho^*, h^*) = L^*(\lambda^*) = \lambda^* D = \frac{D}{d^2} \leq \frac{1}{4}$$

C. *Saddle Point Solutions for $D > \frac{d^2}{4}$*

In the previous section we saw how the adversary can create pdfs h that generate points y where the decision function makes large errors in classification, therefore the idea of using an "indecision" region can help the decision function in regions where deciding between the two hypothesis is prone to errors. In this framework we allow $\rho(y)$ to output $\neg$ when not enough information is given in y in order to decide between $m = 0$ or $m = 1$.

Let $C(i, j)$ represent the cost of deciding for i ($\rho = i$) when the true hypothesis was $m = j$. By using as an evaluation metric the probability of error, we have been so far minimizing the expected cost $\mathbb{E}[C(\rho, m)]$ when $C(0,0) = C(1,1) = 0$ and $C(1,0) = C(0,1) = 1$. We now extend this evaluation metric by incorporating the cost of not making a decision: $C(\neg, 0) = C(\neg, 1) = \alpha$.

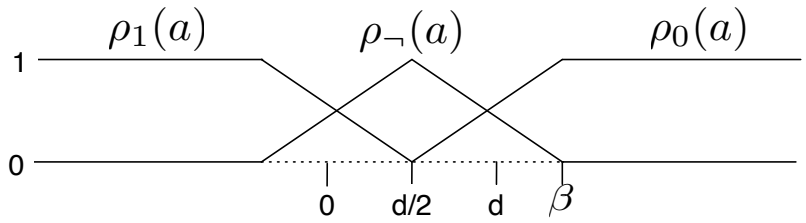


Figure 5.6: $\rho_\neg$ represents a decision stating that we do not possess enough information in order to make a reliable selection between the two hypotheses.

If we let $\alpha < \frac{1}{2}$, it is easy to show (assuming $\Pr[m=0] = \Pr[m=1] = 0.5$) that a decision function that minimizes $\Psi(\rho, h) = \mathbb{E}[C(\rho, m)]$ has the following form:

$$\rho^*(y) = \begin{cases} 1 & \text{if } \frac{f(y|x_1)}{f(y|x_0)} > \frac{1-\alpha}{\alpha} \\ \neg & \text{if } \frac{1-\alpha}{\alpha} > \frac{f(y|x_1)}{f(y|x_0)} > \frac{\alpha}{1-\alpha} \\ 0 & \text{if } \frac{f(y|x_1)}{f(y|x_0)} < \frac{\alpha}{1-\alpha} \end{cases} \tag{5.24}$$

and whenever $\frac{f(y|x_1)}{f(y|x_0)}$ equals either $\frac{\alpha}{1-\alpha}$ or $\frac{1-\alpha}{\alpha}$ the decision is randomized between 1 and $\neg$ and between $\neg$ and 0 (respectively).

Under our non-blind watermarking model the expected cost becomes:

$$\Psi(\rho, h) = \frac{1}{2} \int \{[\rho_1(x+d) + \alpha\rho_\neg(x+d)] + [\rho_0(x) + \alpha\rho_\neg(x)]\} \, h(x)dx$$

where *rho*$_i$ is the probability of deciding for i and where $\rho_0(x) + \rho_\neg(x) + \rho_1(x) = 1$.

Given ρ, the Lagrangian for the optimization problem of the adversary is:

$$L(\lambda, h) = \frac{1}{2} \int \left[\rho_1(a+d) + \alpha\rho_\neg(a+d) + \rho_0(a) + \alpha\rho_\neg(a) - \lambda a^2\right] h(a)da + \lambda D$$

Consider now the decision function given in Figure 5.6. Following the same reasoning as in the previous chapter, it is easy to show that for $\beta = \frac{3}{2}d$, the maximum values for $L(\lambda, h)$ occur for $a = 0$ and $a = \pm d$. The optimal distribution h has the following form:

$$h^*(a) = \frac{p}{2}\delta(-d) + (1-p)\delta(0) + \frac{p}{2}\delta(d)$$

The decision function ρ is Bayes optimal for this attack distribution only if the likelihood ratio for $a = 0$ is equal to $\frac{1-\alpha}{\alpha}$, (i.e., if $\frac{1-p}{p/2} = \frac{1-\alpha}{\alpha}$) and if the likelihood ratio for $a = \pm d$ () is equal to $\frac{\alpha}{1-\alpha}$ (i.e., $\frac{p/2}{1-p} = \frac{\alpha}{1-\alpha}$).

This optimality requirement places a constraint on α: $\alpha = 2p - 1$. Furthermore, the distortion constraint implies the adversary will select $\mathbb{E}[a^2] = pd^2 = D$. Since we need $\alpha < \frac{1}{2}$ in order to make use of the "indecision" region, the above formulation is thus satisfied for $D \leq \frac{3}{4}d^2$.

BIBLIOGRAPHY

[1] C. Kruegel, D. Mutz, W. Robertson, G. Vigna, and R. Kemmerer. Reverse Engineering of Network Signatures. In *Proceedings of the AusCERT Asia Pacific Information Technology Security Conference*, Gold Coast, Australia, May 2005.

[2] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, 1998.

[3] W. Lee, S. J. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the IEEE Symposium on Security & Privacy*, pages 120–132, Oakland, CA, USA, 1999.

[4] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security & Privacy*, pages 133–145, Oakland, CA, USA, May 1999.

[5] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In D. Barbara and S. Jajodia, editors, *Data Mining for Security Applications*. Kluwer, 2002.

[6] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Bayesian event classification for intrusion detection. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC)*, pages 14–24, December 2003.

[7] Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*, pages 1–7, November 1999.

[8] John E. Gaffney and Jacob W. Ulvila. Evaluation of intrusion detectors: A decision theory approach. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 50–61, Oakland, CA, USA, 2001.

[9] Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Skoric. Measuring intrusion detection capability: An information-theoretic approach. In *Proceedings of ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS '06)*, Taipei, Taiwan, March 2006.

[10] Giovanni Di Crescenzo, Abhrajit Ghosh, and Rajesh Talpade. Towards a theory of intrusion detection. In *ESORICS 2005, 10th European Symposium on Research in Computer Security*, pages 267–286, Milan, Italy, September 12-14 2005. Lecture Notes in Computer Science 3679 Springer.

[11] Software for empirical evaluation of IDSs. Available at http://www.cshcn.umd.edu/research/IDSanalyzer.

[12] S. Forrest, S. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for unix processes. In *Proceedigns of the 1996 IEEE Symposium on Security & Privacy*, pages 120–12, Oakland, CA, USA, 1996. IEEE Computer Society Press.

[13] M. Schonlau, W. DuMouchel, W.-H Ju, A. F. Karr, M. Theus, and Y. Vardi. Computer intrusion: Detecting masquerades. Technical Report 95, National Institute of Statistical Sciences, 1999.

[14] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *IEEE Symposium on Security & Privacy*, pages 211–225, Oakland, CA, USA, 2004.

[15] D. J. Marchette. A statistical method for profiling network traffic. In *USENIX Workshop on Intrusion Detection and Network Monitoring*, pages 119–128, 1999.

[16] Salvatore Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Phil Chan. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, pages 130–144, January 2000.

[17] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, 1991.

[18] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech'97)*, pages 1895–1898, Rhodes, Greece, 1997.

[19] N. Japkowicz, editor. *Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets*, 2000.

[20] N. V. Chawla, N. Japkowicz, and A. Kołcz, editors. *Proceedings of the International Conference for Machine Learning Workshop on Learning from Imbalanced Data Sets*, 2003.

[21] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kołz. Editorial: Special issue on learning from imbalanced data sets. *Sigkdd Explorations Newsletter*, 6(1):1–6, June 2004.

[22] Cèsar Ferri, Peter Flach, José Hernández-Orallo, and Nicolas Lachinche, editors. *First Workshop on ROC Analysis in AI*, 2004.

[23] Cèsar Ferri, Nicolas Lachinche, Sofus A. Macskassy, and Alain Rakotomamonjy, editors. *Second Workshop on ROC Analysis in ML*, 2005.

[24] C. Drummond and R. Holte. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207, 2001.

[25] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *DARPA Information Survivability Conference and Exposition*, volume 2, pages 12–26, January 2000.

[26] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, March 2001.

[27] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 2nd edition, 1988.

[28] H. L. Van Trees. *Detection, Estimation and Modulation Theory, Part I*. Wiley, New York, 1968.

[29] Alvaro A. Cárdenas, John S. Baras, and Karl Seamon. A framework for the evaluation of intrusion detection systems. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, California, May 2006.

[30] D. Wagner and P. Soto. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 255–264, Washington D.C., USA, 2002.

[31] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna. Automating mimicry attacks using static binary analysis. In *Proceedings of the 2005 USENIX Security Symposium*, pages 161–176, Baltimore, MD, August 2005.

[32] G. Vigna, W. Robertson, and D. Balzarotti. Testing Network-based Intrusion Detection Signatures Using Mutant Exploits. In *Proceedings of the ACM Conference on Computer and Communication Security (ACM CCS)*, pages 21–30, Washington, DC, October 2004.

[33] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265. ACM Press, 2000.

[34] Y. Zhang, W. Lee, and Y. Huang. Intrusion detection techniques for mobile wireless networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 9(5):545–556, September 2003.

[35] G. Vigna, S. Gwalani, K. Srinivasan, E. Belding-Royer, and R. Kemmerer. An Intrusion Detection Tool for AODV-based Ad Hoc Wireless Networks. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 16–27, Tucson, AZ, December 2004.

[36] Sonja Buchegger and Jean-Yves Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of Tenth Euromicro PDP (Parallel, Distributed and Network-based Processing)*, pages 403 – 410, Gran Canaria, January 2002.

[37] The MIT lincoln labs evaluation data set, DARPA intrusion detection evaluation. Available at http://www.ll.mit.edu/IST/ideval/index.html.

[38] J. McHugh. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by the Lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):262–294, November 2000.

[39] R. E. Blahut. *Principles and Practice of Information Theory*. Addison-Wesley, 1987.

[40] Jean-Pierre Hubaux Maxim Raya and Imad Aad. DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots. In *Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys2004)*, Boston, Massachussets, June 2004.

[41] Svetlana Radosavac, John S. Baras, and Iordanis Koutsopoulos. A framework for MAC protocol misbehavior detection in wireless networks. In *Proceedings of the 4th ACM workshop on Wireless Security (WiSe 05)*, pages 33–42, 2005.

[42] S. Radosavac, G. V. Moustakides, J. S. Baras, and I. Koutsopoulos. An analytic framework for modeling and detecting access layer misbehavior in wireless networks. *submitted to ACM Transactions on Information and System Security (TISSEC)*, 2006.

[43] Pradeep Kyasanur and Nitin Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *Proceedings of the International Conference on Dependable Systems and Networks*, June 2003.

[44] Alvaro A. Cárdenas, Svetlana Radosavac, and John S. Baras. Detection and prevention of mac layer misbehavior in ad hoc networks. In *Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks (SASN 04)*, 2004.

[45] B. E. Brodsky and B. S. Darkhovsky. *Nonparametric Methods in Change-Point Problems*, volume 243 of *Mathematics and Its Applications*. Kluwer Academic Publishers, 1993.

[46] Pierre Moulin and Aleksandar Ivanović. The zero-rate spread-spectrum watermarking game. *IEEE Transactions on Signal Processing*, 51(4):1098–1117, April 2003.