Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE AUDIT GAMES

PRESENTED BY **ARUNESH SINHA**

ACCEPTED BY THE DEPARTMENT OF

Electrical and Computer Engineering

Anupam Datta
_____ _____
ADVISOR, MAJOR PROFESSOR DATE

Jelena Kovačević
_____ _____
DEPARTMENT HEAD DATE

APPROVED BY THE COLLEGE COUNCIL

Vijayakumar Bhagavatula
____ _____ _____
DEAN DATE

Audit Games

Submitted in partial fulfillment for the requirements for
the degreee of
Doctor of Philosophy
in
Electrical and Computer Engineering

Arunesh Sinha
B.Tech, Electrical Engineering, IIT Kharagpur, India

Carnegie Mellon University
Pittsburgh, PA

July, 2014

Dedicated to my wife Pushpa and son Aditya for uncountable reasons.

Abstract

Modern organizations (e.g., hospitals, banks, social networks, search engines) hold large volumes of personal information, and rely heavily on auditing for enforcement of privacy policies. These audit mechanisms combine automated methods with human input to detect and punish violators. Since human audit resources are limited, and often not sufficient to investigate all potential violations, current state-of-the-art audit tools provide heuristics to guide human effort. However, numerous reports of privacy breaches caused by malicious insiders bring to question the effectiveness of these audit mechanisms.

Our thesis is that *effective audit resource allocation* and *punishment levels* can be efficiently computed by modeling the audit process as a game between a rational auditor and a *rational* or *worst-case* auditee. We present several results in support of the thesis. In the worst-case adversary setting, we design a game model taking into account organizational cost of auditing and loss from violations. We propose the notion of *low regret* as a desired audit property and provide a *regret minimizing audit algorithm* that outputs an optimal audit resource allocation strategy. The algorithm improves upon prior regret bounds in the partial information setting. In the rational adversary setting, we enable *punishments* by the auditor, and model the adversary's utility as a trade-off between the benefit from violations and loss due to punishment when detected. Our *Stackelberg game* model generalizes an existing deployed security game model with punishment parameters. It applies to natural auditing settings with multiple auditors where each auditor is restricted to audit a subset of the potential violations. We provide novel polynomial time algorithms to *approximate the non-convex optimization problem* used to compute the Stackelberg equilibrium. The algorithms output optimal audit resource allocation strategy and punishment levels. We also provide a method to reduce the optimization problem size, achieving up to 5x *speedup* for realistic instances of the audit problem, and for the related security game instances.

Acknowledgments

I would like to thank my advisor, Professor Anupam Datta, for his belief in my abilities and for his support of my research interests. Anupam's clarity in thought, dedication and and enthusiasm are inspiring. He taught me, even as a first year graduate student, to think big and aim high. I would like to thank Professor Nicolas Christin and Professor Ariel D. Procaccia for numerous discussions, helpful feedback and constant support towards the work in this dissertation, and also for serving on my thesis committee. I would like to thank Professor Virgil Gligor for the encouragement he provided throughout these years, for valuable feedback about my work, and for serving on my thesis committee. I would like to thank Professor Milind Tambe for valuable feedback about this dissertation, and for serving on my thesis committee.

I am thankful to all of my co-authors, especially Jeremiah Blocki, for all of the great ideas, hard work and late nights. As mentors, Professors Lujo Bauer, Limin Jia and Deepak Garg expanded my horizons and fostered my growth as a researcher. I also benefited immensely from the interactions with Divya Sharma and Michael Tschantz.

I was generously supported by the Bertucci fellowship for which I am grateful. I am also grateful for the research funding awarded to my advisor by Strategic Health IT Advanced Research Projects on Security (SHARPS), National Science Foundation (NSF) and Air Force Office of Scientific Research (AFOSR), which supported me financially in the Ph.D. program.

I am thankful to my parents for providing me with a nurturing environment, and supporting me in all the endeavors of my life. Finally, my most heartfelt thanks go to Pushpa: my wife, my biggest fan and also my harshest critic. Her constant love and encouragement has enabled me to come this far.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Preventive access and information flow control mechanisms form the cornerstone of modern information security and privacy, but are inadequate to enforce desired policies in many practical scenarios. In particular, modern organizations that hold large volumes of personal information (e.g., hospitals, banks, and Web services providers like Google and Facebook) rely heavily on audit mechanisms for enforcement of privacy laws and policies. The importance of auditing in these settings arises from several considerations.

First, privacy policies go beyond access control policies in that they restrict the *use* of personal information to certain *purposes*. For example, the HIPAA Privacy Rule for healthcare privacy in the US restricts use of protected health information to purposes like treatment, payment, and operations. Violations occur when healthcare professionals use information that they have a legitimate right to access for purposes that are not permitted. Recent studies have revealed that numerous policy violations occur in the real world as hospital employees access records of celebrities, family members, and neighbors motivated by general curiosity, financial gain, child custody lawsuits and other considerations [1, 2, 3, 4]. Prior research [5] argues that audit mechanisms are essential for enforcement of purpose restrictions in privacy policies since there is not sufficient information at the time of access to determine whether the policy is being complied with. Furthermore, while automated methods for detecting purpose violations can direct the effort of human auditors, they cannot replace the human auditors since they can produce false positives. Second, in addition to purpose restrictions, privacy policies often permit or deny information

flows based on other *subjective* predicates that cannot be automatically checked. For example, the HIPAA Privacy Rule permits healthcare organizations to share protected health information with law enforcement officials when they *believe* that the data subject was involved in a crime. While large parts of the HIPAA Privacy Rule can be automatically checked, subjective predicates pertaining to such beliefs typically have to be reviewed by human auditors since audit logs do not contain sufficient information to make a conclusive determination about policy compliance [6]. Third, machine learning based tools [7, 8, 9] as well as state-of-the-art healthcare audit tools like FairWarning [10] (that perform SQL queries over audit logs) also produce false positives and require review by human auditors.

Human auditors are a limited resource. They typically cannot inspect every potential violation. Given the resource constraints and lack of quantitative auditing guarantees, auditing tools employ heuristics to direct the effort of human auditors. For example, Fairwarning [10], a popular tool for HIPAA privacy auditing in hospitals, enables auditing all the accesses made to a celebrity's electronic health record (EHR). Following detected violations, hospitals punish the violators using a punishment policy decided by the higher management that may include monetary fines, suspension or even firing from job. An example maybe a $90 cut in paycheck for copying sensitive data to a portable device. Often the punishments vary in degree depending on the severity of the violations. This approach to auditing and punishments is indeed reasonable, because HIPAA violations for celebrities are much more costly for the hospital than a non-celebrity case and punishments should be proportional to the crime. However, the approach is not based on any mathematical models and thus has no optimality guarantees. Would it perhaps be more effective to randomly audit 95% of all accesses to celebrity EHRs and increase the auditing of non-celebrity EHRs? Also, would it perhaps be more effective to impose a fine of $100 for copying sensitive data to a portable device? These questions can be addressed by studying the economics of auditing in an appropriate mathematical model, that in turn yields notions of optimal auditing.

In more detail, the auditor needs to balance auditing costs, potential economic damages due to violations and the economic impact of the punishment policy. Punishment policy has economic consequences for the auditor (organization) since punishment levels affect the productivity of auditees that in turn affects the revenue. The auditees, if rational, will weigh their gain from violating policies against loss

from getting caught by an audit and punished. The actions of one party impact the actions of the other party: if auditees never violate, the auditor does not need to audit; likewise, if the organization never audits, auditees can violate policies in total impunity. Given this strategic interdependency, we model the auditing process as a game between the auditor and auditees. The game is parameterized by quantifiable variables such as the cost of breach, and the cost of auditing, among others. For example, the economic impact of a violation is a combination of *direct and indirect costs*; direct costs include breach notification and remedial cost, and indirect costs include loss of customers and brand value. The 2010 Ponemon Institute report [11] states that the average cost of privacy breach *per record* in health care is $301 with indirect costs about two thirds of that amount. Of course, certain violations may result in much higher direct costs, e.g., in the case of snooping on a celebrity's records in Kaiser Permanente's Bellflower hospital in the state of California, cost were $\$25,000$ per record (up to $\$250,000$ in total) in fines alone. [3].

More generally, our thesis is as follows:

> Effective audit resource allocation and punishment levels can be efficiently computed by modeling the audit process as a game between a rational auditor and a rational or worst-case auditee.

Our results in support of the thesis explore different assumptions about the rationality of the auditee, notions of audit optimality, and algorithms for efficiently computing optimal resource allocation and punishment levels. We present a brief overview of our results in the following sections. Finally, the generality of our economic model of auditing enables its application to areas beyond information security and privacy. For instance, traffic inspections coupled with fines, ticket inspections in rail systems and financial audits are all audit scenarios that are instances of our model.

1.1 The Worst-case Adversary

Our first cut at modeling the audit process explores what could be done without restricting the adversary's (auditee's) behavior in any manner. We model the adversary as an agent that performs certain tasks (e.g., accesses to EHR), and a subset of these tasks are policy violations. The auditor inspects a subset of the tasks, and

detects violations from among the inspected set. As different types of violations may have a different impact on the organization, we consider the auditee's action as a vector of tasks and the auditor's action as a vector of inspections with each component of the vector denoting a particular type. For example, in a hospital there could be two types: celebrity record accesses and non-celebrity record accesses. The auditor can choose to inspect accesses of a certain type lightly, moderately, or heavily, thus, the auditor has 9 possible pairs of inspections.

Next, we talk about the utilities for each type; the total utility is a sum of the utilities of each type. The auditor pays an audit cost that increases monotonically with the number of inspections, and it suffers a loss due to violations. The loss also increases monotonically with the number of violations. Further, the loss from externally detected violations is higher than the loss from internally detected violations. Thus, the auditor has an incentive to perform more inspections in order to detect violations on its own. Governmental audits, whistle-blowing, patient complaints [12, 13] are all examples of situations that could lead to external detection of violations. Externally detected violations usually cause more economic damage to the organization than internally caught violations. The 2011 Ponemon Institute report [14] states that patients whose privacy has been violated are more likely to leave (and possibly sue) a hospital if they discover the violation on their own than if the hospital detects the violation and proactively notifies the patient. The auditor-auditee interaction is repeated in every audit cycle, which we capture by modeling the interaction as a repeated game.

It may seem impossible to provide any guarantee with a worst-case adversary in a post-hoc inspection setting. However, we use the concept of *regret* from game theory [15] to provide a desirable mathematical property of auditing in such a model. Informally, low regret of a strategy S captures the notion that S performs as well as any strategy in a given set of strategies fixed prior to the start of the game. The performance of a strategy is measured by the average utility obtained by playing the strategy in T rounds. Hence, regret for T rounds is the difference is average utility after T rounds, with low regret meaning that the regret is not more than zero as T goes to infinity. We propose low regret as the desirable property of an audit mechanism in this setting. Continuing our hospital example, the auditor has 9 actions in each round, and the set of strategies she considers is to play the same 9 actions in every round. Then, a low regret strategy S would guarantee an average

utility that approaches the best average utility that any of the 9 strategies provides.

A *regret minimizing algorithm* computes a strategy that achieves low regret with respect to any given finite set of strategies of cardinality N. There are a number of regret minimizing algorithms proposed in the learning theory literature [15]. A measure of the quality of any regret minimization algorithm is the rate at which regret converges to zero. The best regret minimization algorithm in the basic setting, named multiplicative weight updates [16], provides a regret bound of $O\left(\sqrt{\frac{\log N}{T}}\right)$, where T is the number of rounds.

Our model presents additional challenges beyond the basic regret minimization setting; these are imperfect information (also called partial information) and sleeping experts. Imperfect information refers to the scenario where the adversary's action is not observed for sure; instead some signal of that action is observed. This is the case in our model also, as the number of violations may not be perfectly observed, since only a subset of the accesses are inspected. Sleeping experts refers to the scenario where the action space of the defender may change in different rounds of the game. Indeed, due to the possibly different budgetary constraints imposed exogenously, the maximum number of inspections that the auditor can perform can differ from round to round. Both these problems of imperfect information [15] and sleeping experts [17] have been addressed in the literature. The best regret bounds in such cases is $O\left(\frac{N^{1/3}\log N}{T^{1/3}}\right)$.

Main Result. We build on existing algorithms to propose *Regret Minimizing Audits* (RMA) [18], a mechanism to achieve low regret in our model. RMA improves upon existing algorithms obtaining regret bounds of $O\left(\sqrt{\frac{\log N}{T}}\right)$, under the mild assumption that we can obtain an unbiased estimator of the actual action from the observed signal about the action. We discuss the details of the model, algorithm and assumptions in Chapter 3.

1.2 The Rational Adversary

A natural intuition is that stronger guarantees could be provided if we constrain the adversary's behavior. Following this intuition, we propose an audit model with rational adversaries. Building on our model with arbitrary adversary, we model the

payoff for the adversary as the sum of benefit gained from committing a violation and the loss from punishment if the violation is detected. The benefit is quantifiable using information from existing studies or by human judgment. For example, reports [19, 20] indicate that on average the *personal benefit* of a hospital employee from selling a common person's health record is $50. As punishments do affect the behavior of the adversary, it is critical for the auditor to choose the right level of punishment. As a consequence, we add the choice of a punishment level to the action space of the auditor.

However, punishment is not free for the auditor—the intuition being that a high punishment level creates a hostile work environment, leading to lack in productivity of employees that results in loss for the organization (auditor). As a consequence, the auditor cannot impose infinite punishment and deter any adversary. We model the auditor's cost for a punishment level as a loss proportional to the choice of the punishment level. The auditor moves first by committing to an inspection and punishment strategy, followed by the best response of the adversary. Our model results in a game that generalizes security games [21] with punishments. Security games is an area of extensive research in the AI community, that has also found real life application in airport security, airline security, etc. Similar to security games we use the concept of Stackelberg equilibrium in our game. However, distinct from that work, the punishment parameter in our model makes the equilibrium computation a non-convex optimization problem. Non-convex optimization is known to be hard to even approximate in the general case [22].

Single defender resource. As a first cut, we demonstrate a fully polynomial time approximation scheme (FPTAS) for a simple rational model where the auditor has just one inspection, and the adversary commits one violation [23]. Following terminology of security games, we call each task of the adversary a target, and a violation as an attack on the target. Our approach to computing the Stackelberg equilibrium is based on the multiple LPs technique of Conitzer and Sandholm [24]. Assume the targets are $t_1, \ldots, t_n$. The defender solves multiple optimization problem that maximizes her utility, one each with the constraint that target t_i is attacked. Then, the defender chooses the best solution from among these n maximizers. Mathematically, for each optimization problem, the defender chooses $p_1, \ldots, p_n$ (probability of auditing target t_i) and x (punishment level) in order to maximize her utility under

the constraint that the adversary attacks target t_i. The adversary attacking target t_i is mathematically stated as the utility of the adversary in attacking target t_i is not less than the utility in attacking any other target (these adversary utilities depends on $p_1, \ldots, p_n, x$).

The main idea in the FPTAS optimization is the property that at the optimal point there is a subset of targets such that all targets in the subset are most (and equally) attractive for the adversary and are audited by the defender, and all the other targets are not audited at all. Further, the number of such subsets of targets that we need to consider is just linear in the number of targets.

Then, iterating over each such subset, solving the resultant sub-problem and taking the best solution from all the sub-problems provides a global best solution. We show that it is possible to reduce each sub-problem to a two variable optimization problem. Next, the two variable optimization problem can be reduced to finding the roots of single variable polynomials. The roots of a polynomial can be approximated efficiently using methods of Schonage [25]. Hence, the overall algorithm runs efficiently providing an approximate algorithm. The details of the model and algorithm is presented in Chapter 4.

Multiple defender resources. We extend the result to the more realistic scenario of multiple inspections with constraints on the targets that each inspection can inspect. We treat each available inspection as an resource, thereby, imposing the natural constraint that each resource can inspect at most one target. Further, some special types of targets may only be inspected by specialized human auditors, which imposes constraints on the set of targets that each resource can potentially inspect. Examples of specialized auditing include scenarios such as billing related case investigated by financial auditors from billing department, localized auditing in which managers audit their direct reportees and the same localized auditing, but with company wide auditors for cases that some mangers are not capable of auditing. Mathematically, we introduce variable p_i^j, the probability of inspection resource j inspecting target i. Some of these variables are forced to be zero, reflecting the specialized auditing scenario. Then, the probability of inspecting a target t_i is $p_i = \sum_j p_i^j$.

We first provide an intuitive fixed parameter tractable (FPT) solution, obtained by discretizing the punishment parameter x and solving the resultant linear pro-

grams for every fixed punishment level. The FPT can also handle multiple (constant) number of violations by the adversary. Then, we present an approach to reduce the problem size by eliminating many variables of the optimization problem, namely all the p_i^j's, though at the cost of increasing the number of constraints (in p_i's and x) in the problem. We characterize the conditions under which the number of constraints do not increase exponentially, and hence allow for solving the audit game problem efficiently (FPTAS). These conditions capture many specialized auditing scenarios including all the specialized auditing examples we listed above.

In more detail, given polynomially many number of constraints in p_i's and x, we can reduce the problem to $n-1$ number of sub-problems using the exact technique used in the single defender resource case. Then, building on the approach in the single defender resource case, each sub-problem can be reduced to a two variable optimization problem, which can be further be reduced to finding roots of polynomials. As stated earlier, the roots of a polynomial can be approximated efficiently using methods of Schonage [25]. The approach of reducing the problem size also helps in greatly reducing the equilibrium computation time in the analogous security games scenario, and in the FPT approach for audit games. We demonstrate this improvement in running time experimentally for the FPT approach, achieving improvement of 5x in running time with 200 targets, 10 human auditors with 10 inspections each (100 inspection resources) in a localized auditing structure, where each auditor inspects 20 targets.

Target specific punishments. Finally, we extend the model to the scenario where an auditor can choose a different punishment level $(x_1, \ldots, x_n)$ for different targets $t_1, \ldots, t_n$. This extension to the model reflects the common practice in law of proportional punishment [26]. The naive approach of discretizing each $x_1, \ldots, x_n$ does not yield a FPT algorithm. We provide a FPT for this scenario by discretizing only p_i, x_i (for the case when t_i is the target under attack) and reducing the resultant optimization problem to a second order cone program (SOCP), that is known to be solvable efficiently. The details of the extension are presented in Chapter 5.

1.3 Repeated Rational Audit Games

As stated earlier, the auditor-auditee interaction is repeated in every audit cycle. We prove that repeating the one round Stackelberg commitment of the auditor results in a *sub-game perfect equilibrium* of the repeated game. Sub-game perfect equilibrium is a refinement of Nash equilibrium that is considered apt for repeated games.

Further, relaxing the rationality assumption for the adversary, we propose a near-rational model of the adversary. In the near-rational model the adversary plays his best response, except, she may take any other action with a small probability. This models an adversary with *trembling hand* [27].

We define an *asymmetric approximate* sub-game perfect equilibrium , and prove that repeating the Stackelberg equilibrium from each round with a near-rational adversary leads to such an equilibrium of the game. The asymmetry in the equilibrium is particularly relevant for security settings. Approximate equilibrium in game theory allows players to deviate from the equilibrium action as long as they achieve their equilibrium payoff within an additive factor of ϵ. In our asymmetric version, we restrict the adversary to $\epsilon = 0$, i.e., the rational adversary is not allowed to deviate at all. The defender is still allowed to deviate within approximation factor ϵ. It is important to disincentivize the adversary from deviating, or else both players deviating could result in huge loss for the defender.

Finally, we demonstrate that our model can explain phenomena that happen in real world auditing scenarios, and also provide informed suggestions on policies that could encourage organizations to perform more auditing. For example, one way to encourage more auditing is to increase the probability of external detection and the cost of externally detected violations, possibly by more external audits and higher fines for lax internal auditing. An interesting counterintuitive prediction of our model is that increasing both the cost of external detected violation and internally detected violations equally may not incentivize organizations to audit more. An example of increasing cost of internally detected violations is mandating that organizations send out notification to affected parties in case of a breach. While such laws appear effective, our model predicts that they will not lead to better auditing practice by organizations.

1.4 Scope of Work

This dissertation aims to provide techniques that enable organizations to audit in a cost-optimal manner. We emphasize that cost-optimality for the organization may not align with the goal of deterring the employees from committing violations. Indeed, as we demonstrate in Chapter 6.4, if its too costly to deter an employee then the cost-optimal strategy is to not punish the employee at all. However, our techniques can be easily extended to change the goal from cost-optimality to deterrence, an example of which is presented in Chapter 4.2 where we use a *dummy target* to symbolize no attack and solve the scenario when the *dummy target* is attacked.

While deterrence is a desirable property of auditing, the natural goal of a rational organization is cost-optimality. Our focus on cost-optimality captures phenomenon that occur in auditing in the real world; in the process enabling the study of interventions that encourage deterrence. We discuss some examples of possible interventions in Chapter 6.4, such as increased external audit of organizations and increased fines for organizations.

Also, our abstraction contains a number of assumptions, some of which could be removed easily while relaxing other assumptions requires solving hard research problems. These assumptions include abstractions such as a violation is caught for sure when inspected, every violation is independent of other violations and adversaries do not collude. We discuss these assumptions in the Chapter 7.1.

1.5 Summary of Contributions

The main contributions of this dissertation are as follows:

- Identifying regret as a notion of optimality in the worst-case adversary audit setting.
- A regret minimizing audit algorithm with regret bound $O\left(\sqrt{\frac{\log N}{T}}\right)$, which is better than known regret bounds in the imperfect (partial) information setting [15].
- A leader-follower game model of auditing with rational adversaries that generalizes a standard security game model [21] with a punishment level parameter.
- Efficient algorithms for computing the Stackelberg equilibrium in the rational

setting. In particular, we provide novel ways of obtaining FPTAS for particular instances of non-convex optimization problems and also provide novel ways of improving the running time for the FPT algorithm for the same instances. We achieve a 5x improvement for instances with 200 targets and 100 inspection resources.

- An asymmetric notion of approximate sub-game perfect equilibrium in games with particular relevance for security. This notions allows the defender to deviate from equilibrium with an ϵ additive factor bound on the difference from optimal utility considering fixed equilibrium play by the adversary. However, it restricts the adversary to not deviate at all. This is important, because deviation by the adversary could result in significant loss for the defender.
- Policy guidelines on the steps that could be taken to provably encourage organizations to audit better. For example, increased external inspection of organization coupled with appropriate fines will encourage organizations to audit more effectively.

Chapter 2

Background and Related Work

2.1 Game Theory

Game theory is an area of study that seeks to model strategic interaction, and predict outcomes of such interaction. Seminal work by John von Neumann [28] and John Nash [29] in 1940's laid the foundations of modern game theory, which has been followed by extensive work on game models in the economics community and more recently, on game models and algorithmic aspects in the computer science community. We provide a basic overview of game theory here; the interested reader can find much more details in books dedicated to game theory [27].

We restrict our attention to two players games, call them P_a, P_b. Each player has an action space given by sets A, B. Actions are the choices available to the players. Each player also has a utility function U_a, U_b, and the signature of both functions is $U : A \times B \to \mathbb{R}$. Note that the dependence of ones utility on the action of the other player, differentiates this set-up from decision theory. Every player is assumed rational, i.e., all players prefer higher utility for themselves. An action profile (a, b) is a pair of actions, one taken by each player ($(a, b) \in A \times B$). An action profile may be mixed, i.e., a player may choose a probability distribution over actions. Let ΔX denote all probability distributions over set X, then a mixed action profile is $(a, b) \in \Delta A \times \Delta B$. The utilities with mixed actions are expected values over the distribution of actions.

For sake of completeness, we define the Nash Equilibrium (NE) for a simultaneous move game, but, we do not use NE in this dissertation. An (mixed) action profile

(a^*, b^*) is called a (mixed) NE if

$$\forall a'.\ U_a(a^*, b^*) \geq U_a(a', b^*) \ \text{ and } \ \forall b'.\ U_b(a^*, b^*) \geq U_b(a^*, b') \ .$$

A game maybe simultaneous move game, i.e., players move simultaneously, or a sequential move game, i.e., players move in a sequence. The one round sequential move game is called a Stackelberg game. This game involves a leader moving first, followed by the follower playing a best response. The best response of follower P_b is defined as the function br such that

$$\forall a, br'.\ U_b(a, br(a)) \geq U_b(a, br'(a)) \ .$$

Observe that the best response function may not be unique, since, there can be two action of P_b such that both are best responses to a given action of P_a. Best response functions have an equivalent definition as above in case of mixed action a^*, except the utilities are expectations taken over the distribution of a^*. Since the leader moves first, a particular function br automatically chooses the action of the follower. Thus, the action of the P_b can be regarded as choice of function br. Then, (a^*, br) (a^* could be mixed) is the Stackelberg equilibrium in a sequential move game if

$$\forall a'.\ U_a(a^*, br(a^*)) \geq U_a(a', br(a')) \ .$$

As a result of the completely different dynamics in sequential games, the set of Stackelberg equilibrium may be different from the set of NE in the corresponding simultaneous move game. The following example makes this clear.

		Player P_b	
		b_1	b_2
Player P_a	a_1	4, 2	2, 1
	a_2	5, 2	3, 3

The NE in this game is (a_2, b_2), and, the Stackelberg equilibrium is (a_1, b_1).

Adding another condition on the Stackelberg equilibrium—that the follower breaks ties in favor of the leader—yields a strong Stackelberg equilibrium. We refer the reader to a paper by Korzhyk et al. [30] for details of strong Stackelberg equilibrium, including the result that the leader always obtains better utility in a strong Stackelberg equilibrium than the utility in a NE of the simultaneous move game. Another point to note is that we consider best responses to be pure actions (rather than mixed), and this can be done without loss of generality as any action in the support of the mixed action is by itself a best response. The details of Stackelberg equilibrium and its computation can be found in Conitzer et al. [24], which we elaborate on in the next section.

2.2 Stackelberg Equilibrium Computation and Optimization

In a Stackelberg equilibrium, the best response of the followers induces a partition of the action space of the leader, such that in each partition the best response of the adversary is same. Given finite action spaces, it is possible to find the best leader action in each such partition by solving an optimization problem. This optimization problem maximizes the leader's utility, subject to the constraint that a given action is the best response of the follower. This it the multiple optimization approach. The complexity of this approach for the basic leader-follower game and Bayesian Stackeleberg games was studied in a seminal paper by Conitzer and Sandholm [24]. They demonstrate that for a two player game, the Stackelberg equilibrium can be computed efficiently for bi-matrix games using linear programs.

However, under additional constraints on the game, computing Stackelberg equilibrium is not easy. The computation of Stackelberg equilibrium in security games [21] is not easy in many cases. Security games has a model similar to our audit games model, but, without punishments. However, security games has many variations not explored in audit games yet, such a resource inspecting multiple targets simultaneously. These aspects make the computation of Stackelberg equilibrium hard in many instances of security games [31].

However, distinct from the bi-matrix games considered in that paper, our scenario involves an action that is continuous and the utility function of the following in

non-convex in the actions of the leader. These considerations make our optimization problem non-convex.

While linear programs and many instance of convex programs can be solved in polynomial times using interior point methods [32], it is known that non-convex optimization is even hard to approximate [22]. However, there are some instances of non-convex optimization problems that have been solved or approximated efficiently [22], such as concave objectives with low rank (see the definition of low rank in [22]). Also, many apparently non-linear convex programs, such as semi-definite and second order cone programs [32], are linear programs in disguise and can be solved extremely efficiently.

2.3 Regret Minimization

Regret (referred to as external regret in learning theory) was proposed way back in 1950's [33, 34], along with algorithm that minimized regret. We define regret here in the discrete case relying on the definition in the book Algorithmic game theory [15].

Consider a game played by P_a and P_b. Consider a strategy employed by player P_a which we refer to as Alg. Alg outputs an action to be played by P_a in every round t. The action output maybe a mixed action. Alg depends on the whole history of actions played and payoffs received by P_a. Consider other strategies $s \in S$, that also outputs action to be played by P_a in every round t. Regret of Alg with respect to s is defined as the average difference in expected utility

$$Regret(\mathsf{Alg}, s, T) = \frac{1}{T}\sum_{t=1}^{T} U_a(s(t), \vec{b}_t) - U_a(\mathsf{Alg}(t), \vec{b}_t)$$

where $s(t)$ denotes the action s output in round t, and $\vec{b}_t$ is the t^{th} component of the length T vector $\vec{b}$ which provides the actions played by the other player in the T rounds. This is also the perfect information model, since the actions of the P_b are observed by P_a; perfect information coupled with the knowledge of the game matrix allows P_a to figure out the payoff had it played any other action. We can extend the above definition with respect to a set of strategies S as

$$Regret(\mathsf{Alg}, S, T) = \max_{s \in S} Regret(\mathsf{Alg}, s, T)$$

Alg is a regret minimization algorithm if it guarantees that for any sequence $\vec{b}$ and any set S of bounded size

$$\lim_{T\to\infty} Regret(\mathsf{Alg}, S, T) \to 0$$

Thus, a regret minimization algorithm Alg is a randomized algorithm for playing in a repeated game. The weighted majority algorithm [16] is a regret minimization algorithm. The weighted majority maintains weights w_s for each of the $|S|$ fixed strategies of the defender. w_s^t is the weight of the strategies before round t has been played. The weights determine a probability distribution over actions, p_s^t denotes the probability of playing s at time t. In any given round the algorithm attempts to learn the optimal distribution over actions by increasing the weights of experts that performed better than its current distribution and decreasing the weights of experts that performed worse. This algorithm obtains a bound on regret of $O\left(\sqrt{\frac{\log N}{T}}\right)$, which goes to 0 as T goes to infinity.

An extension of the above scenario is when the action b of the other player is not observed; such settings are called imperfect information or partial information. It is possible to perform regret minimization in this setting also [15], by using the perfect information algorithm as a sub-routine. However, the convergence rate of regret to 0 becomes worse, with a rate of $O\left(\frac{N^{1/3}\log N}{T^{1/3}}\right)$ achieved in [35].

2.4 Auditing and Risk Management

There are different directions explored in research on auditing. A line of work in computer security uses evidence recorded in audit logs to understand why access was granted and to revise access control policies if unintended accesses are detected [36, 37, 38]. In contrast, we use audits to detect violations of policies, such as those restricting information use to specified purposes, that cannot be enforced using access control mechanisms.

Cederquist et al. [39] present logical methods for enforcing a class of policies, which cannot be enforced using preventive access control mechanisms, based on evidence recorded on audit logs. The evidence demonstrating policy compliance is presented to the auditor in the form of a proof in a logic and can be checked

mechanically. In contrast, our focus is on policies that cannot be mechanically enforced in their entirety, but require involvement of human auditors. In addition, the new challenges in our setting arises from the imperfect nature of audits.

Garg et al. [6] present an algorithm that mechanically enforces objective parts of privacy policies like HIPAA based on evidence recorded in audit logs and outputs subjective predicates (such as beliefs) that have to be checked by human auditors. This work provides evidence of our claim that human intervention is necessary in auditing. In the same vein, machine learning based tools for auditing [8, 9] suffer from false positives and negatives, requiring human intervention in auditing.

Zhao et al. [40] recognize that rigid access control can cause loss in productivity in certain types of organizations. They propose an access control regime that allows all access requests, but marks accesses not permitted by the policy for posthoc audit coupled with punishments for violating policy. They assume specific utility function for the organization and the employees and use a single shot sequential game to analyze the optimal behavior of the players, obtaining closed-form solutions for the audit strategy. Our approach of using a permissive access control policy coupled with audits is a similar idea. However, we also consider a worst-case adversary (employee) because we believe that it may be difficult to identify the exact incentives of the employee in scenarios such as outsourced work. For the rational case, our results apply to any utility function and we present an efficient algorithm for computing the audit strategy. Finally, we restrict the amount of audit inspections because of budgetary constraints. Thus, our game model is significantly more realistic than the model of Zhao et al. [40].

Cheng et al. [41, 42] also start from the observation that rigid access control is not desirable in many contexts. They propose a risk-based access control approach. Specifically, they allocate a risk budget to each agent, estimate the risk of allowing an access request, and permit an agent to access a resource if she can pay for the estimated risk of access from her budget. Further, they use metaheuristics such as genetic programming to dynamically change the security policy, i.e. change the risk associated with accesses dynamically. Our approach to the problem is fundamentally different: we use a form of risk-based auditing instead of risk-based access control. Also, genetic programming is a metaheuristic, which is known to perform well empirically, but does not have theoretical guarantees [43]. In contrast, we provide mechanisms with provable guarantees. Indeed an interesting topic for

future work is to investigate the use of learning-theoretic techniques to dynamically adjust the risk associated with accesses in a principled manner.

Feigenbaum et al. [44] report work in progress on formal definitions of accountability capturing the idea that violators are punished with or without identification and mediation with non-zero probability, and punishments are determined based on an understanding of "typical" utility functions. Operational considerations of how to design an accountability mechanism that effectively manages organizational risk is not central to their work.

Our work is an instance of a risk management technique [41, 45] in the context of auditing and accountability. As far as we know, our technique is the first instance of managing risk in auditing using a game formalism. Risk assessment has been extensively used in many areas [46, 47]; the report by American National Standards Institute [20] provides a risk assessment mechanism for healthcare. Our model also models data breaches that happen due to insider attacks. Reputation has been used to study insider attacks in non-cooperative repeated games [48]; we differ from that work in that the employer-employee interaction is essentially cooperative. Also, the primary purpose of interaction between employer and employee is to accomplish some task (e.g., provide medical care). Privacy is typically a secondary concern. Our model captures this reality by considering the effect of non-audit interactions in parameters like P_f. There are quite a few empirical studies on data breaches and insider attacks [11, 12, 49] and qualitative models of insider attacks [50]. We use these studies to estimate parameters and evaluate the predictions of our model.

2.5 Security Games

Our rational game model is closely related to security games [21]. Security games is a model of physical security and has found application in the real world in scheduling of patrols in LAX airport, scheduling of air marshals in flight and scheduling coast guard patrols. There are many papers (see, e.g., [31, 51, 52]) on security games, and our model adds the additional continuous punishment parameter.

The basic security game model involves a defender with resource constraints defending a large set of targets from an attacker. The attacker is rational and attacks one target, specifically, the target that is most attractive in terms of cost of attack

and gain from attack. The problem becomes challenging with combinatorial constraints on the use of resources. For instance, different resources may be restricted to defending different subsets of resources—called schedules in that literature. Schedules could be singletons, implying each resource defends maximum of one target. Further, resource may be constrained in the schedules they can take up for defense; resources are homogeneous if they can take up any schedule.

In our most general model of rational audit games, we treat each inspection as a resource, and hence singleton schedules arises naturally in our setting. Further, as different auditors may be equipped to inspect different targets, our problem has heterogeneous schedules. As stated earlier, our main contributions include recognizing that punishments should be chosen judiciously to maximize ones owns benefit. The technical problem in audit games is more challenging as it involves non-convex optimization; we rely on specific properties of this problem to obtain a FPTAS. Further, we explore variations specific to audit games such as different punishment for different targets.

There are many approaches to solving security games. One approach [21] is to formulate the problem as a mixed integer linear program (MILP) and employ heuristics such as column generation to solve large instances of the problem. Along the same lines, another paper [53] employs the same approach to solve just for coverage probabilities (the probability of defending a resource), and demonstrating the decomposition into equivalent mixed strategy for the unconstrained case (no scheduling constraints). For the constrained case, the solution only solves for coverage probabilities and does not provide for decomposing the coverage probabilities into mixed strategy. In contrast, our approach transforms the audit game (applicable to security games also) problem with singleton heterogeneous schedules into an equivalent problem with additional linear constraints on the coverage probabilities, such that the coverage probabilities obtained can always be decomposed into mixed strategies. We provide a theoretical proof of the correctness of the above transformation. We also identify conditions in which the number of additional linear constraints are polynomial in number, and demonstrate that in the worst case the number of constraints are exponential.

Another approach [31] focuses on the computational complexity of the problem and demonstrates the complexity of the problem for different types of schedules. In particular, they demonstrate a polynomial time approach to solving the scenario of

singleton heterogeneous schedules. Our approach studies the complexity of solving audit games. However, in contrast to solving for the the probability of each resources inspecting each target, we transform the problem by introducing additional linear constraints on the coverage probabilities, and then solve a system of linear equations to obtain the probability of each resource inspecting each target. This is required to obtain a polynomial time solution for audit games because of the non-convexity of the optimization problem in audit games. We demonstrate that audit games have a polynomial time solution when the number of new linear constraints is polynomial. Moreover, we show that the transformation improves the running time for solving audit games using the FPT approach, and hence also improves the running time for related (singleton, heterogeneous schedules) security games.

Overall, our approach of transforming the problem to a problem with only coverage probabilities in audit games provides for a speed-up for solving security games for singleton heterogeneous schedules. Our approach is also accompanied by theoretical proofs of correctness.

Chapter 3

Regret Minimizing Audits

A formal study of privacy regulations [54] shows that a large fraction of clauses in the HIPAA Privacy Rule [55] requires some input from human auditors to enforce. We seek to develop an appropriate mathematical model for studying audit mechanisms involving human auditors, in the worst case scenario of arbitrary adversaries. Specifically, the model should capture important characteristics of practical audit mechanisms (e.g., the periodic nature of audits), and economic considerations (e.g., cost of employing human auditors, brand name erosion and other losses from policy violations) that influence the coverage and frequency of audits.

This chapter presents the first principled learning-theoretic foundation for audits of this form. Our first contribution is a *repeated game model* that captures the interaction between the defender (e.g., hospital auditors) and the adversary (e.g., hospital employees). The model includes a budget that constrains the number of actions that the defender can inspect thus reflecting the imperfect nature of audit-based enforcement, and a loss function that captures the economic impact of detected and missed violations on the organization. We assume that the adversary is worst-case as is standard in other areas of computer security. We also formulate a desirable property of the audit mechanism in this model based on the concept of *regret* in learning theory [15]. Our second contribution is a novel *audit mechanism* that provably minimizes regret for the defender. The mechanism learns from experience and provides operational guidance to the human auditor about which accesses to inspect and how many of the accesses to inspect. The regret bound is significantly better than prior results in the learning literature. We start with an

overview of our results before describing the model and algorithm in details.

3.1 Overview of Results

Mirroring the periodic nature of audits in practice, we use a repeated game model [27] that proceeds in rounds. A round represents an audit cycle and, depending on the application scenario, could be a day, a week or even a quarter.

Adversary model: In each round, the adversary performs a set of actions (e.g., accesses patient records) of which a subset violates policy. Actions are classified into types. For example, accessing celebrity records could be a different type of action from accessing non-celebrity records. The adversary capabilities are defined by parameters that impose upper bounds on the number of actions of each type that she can perform in any round. We place no additional restrictions on the adversary's behavior. In particular, we do not assume that the adversary violates policy following a fixed probability distribution; nor do we assume that she is rational. Furthermore, we assume that the adversary knows the defender's strategy (audit mechanism) and can adapt her strategy accordingly.

Defender model: In each round, the defender inspects a subset of actions of each type performed by the adversary. The defender has to take two competing factors into account. First, inspections incur cost. The defender has an audit budget that imposes upper bounds on how many actions of each type she can inspect. We assume that the cost of inspection increases linearly with the number of inspections. So, if the defender inspects fewer actions, she incurs lower cost. Note that, because the defender cannot know with certainty whether the actions not inspected were malicious or benign, this is a game of imperfect information [35]. Second, the defender suffers a loss in reputation for detected violations. The loss is higher for violations that are detected externally (e.g., by an Health and Human Services audit, or because information leaked as a result of the violation is publicized by the media) than those that are caught by the defender's audit mechanism, thus incentivizing the defender to inspect more actions.

In addition, the loss incurred from a detected violation depends on the type of violation. For example, inappropriate access of celebrities' patient records might cause higher loss to a hospital than inappropriate access of other patients' records.

Also, to account for the evolution of public memory, we assume that violations detected in recent rounds cause greater loss than those detected in rounds farther in the past. The defender's audit mechanism has to take all these considerations into account in prescribing the number of actions of each type that should be inspected in a given round, keeping in mind that the defender is playing against the powerful strategic adversary described earlier.

Note that for adequate privacy protection, the economic and legal structure has to ensure that it is in the best interests of the organization to invest significant effort into auditing. Our abstraction of the reputation loss from policy violations that incentivizes organizations to audit can, in practice, be achieved through penalties imposed by government audits as well as through market forces, such as brand name erosion and lawsuits.

Regret property: We formulate a desirable property for the audit mechanism by adopting the concept of regret from online learning theory. The idea is to compare the loss incurred when the real defender plays according to the strategy prescribed by the audit mechanism to the loss incurred by a hypothetical defender with perfect knowledge of the number of violations of each type in each round. The hypothetical defender is allowed to pick a fixed strategy to play in each round that prescribes how many actions of each type to inspect. The *regret* of the real defender in hindsight is the difference between the loss of the hypothetical defender and the actual loss of the real defender averaged over all rounds of game play. We require that the regret of the audit mechanism quickly converge to a small value and, in particular, that it tends to zero as the number of rounds tends to infinity.

Intuitively, this definition captures the idea that although the defender does not know in advance how to allocate her audit budget to inspect different types of accesses (e.g., celebrity record accesses vs. non-celebrity record accesses), the recommendations from the audit mechanism should have the desirable property that over time the budget allocation comes close to the optimal fixed allocation. For example, if the best strategy is to allocate 40% of the budget to inspect celebrity accesses and 60% to non-celebrity accesses, then the algorithm should quickly converge towards these values.

Audit mechanism: We develop a new audit mechanism that provably minimizes regret for the defender. The algorithm, which we name RMA (for Regret Minimizing Audit) is efficient and can be used in practice. In each round of the game, the

algorithm prescribes how many actions of each type the defender should inspect. It does so by maintaining weights for each possible defender action and picking an action with probability proportional to the weight of that action. The weights are updated based on a loss estimation function, which is computed from the observed loss in each round. Intuitively, the algorithm learns the optimal distribution over actions by increasing the weights of actions that yielded better payoff than the expected payoff of the current distribution and decreasing the weight of actions that yielded worse payoff.

Our main technical result (Theorem 1) is that the exact bound on regret for RMA is approximately $2\sqrt{2\frac{\ln N}{T}}$ where N is the number of possible defender actions and T is the number of rounds (audit cycles). This bound improves the best known bounds of $O\left(\frac{N^{1/3}\log N}{\sqrt[3]{T}}\right)$ for regret minimization over games of imperfect information. The main novelty is in the way we use a loss estimation function and characterize its properties to achieve the significantly better bounds. Specifically, RMA follows the structure of a regret minimization algorithm for perfect information games, but uses the estimated loss instead of the true loss to update the weights in each round. We define two properties of the loss estimation function—*accuracy* (capturing the idea that the expected error in loss estimation in each round is zero) and *independence* (capturing the idea that errors in loss estimation in each round are independent of the errors in other rounds)—and prove that any loss estimation function that satisfies these properties results in regret that is close to the regret from using an actual loss function. Thus, our bounds are of the same order as regret bounds for perfect information games. The better bounds are important from a practical standpoint because they imply that the algorithm converges to the optimal fixed strategy much faster.

The rest of the chapter is organized as follows. Section 3.2 presents the game model formally. Section 3.3 presents the audit mechanism and the theorem showing that the audit mechanism provably minimizes regret. Section 3.4 discusses the implications and limitations of these results. Section 3.5 describes in detail the loss estimation function, a core piece of the audit mechanism. Section 3.6 presents the outline of the proof of the main theorem of the paper (Theorem 1) while the complete proofs are presented in the appendices.

3.2 Model

We model the internal audit process as a repeated game played between a defender (organization) and an adversary (employees). In the presentation of the model we will use the following notations:

- Vectors are represented with an arrow on top, e.g., $\vec{v}$ is a vector. The i^{th} component of a vector is given by $\vec{v}[i]$. $\vec{v} \leq \vec{a}$ means that both vectors have the same number of components and for any component i, $\vec{v}[i] \leq \vec{a}[i]$.
- Random variables are represented in boldface, e.g., $\mathbf{x}$ and $\mathbf{X}$ are random variables.

The repeated game we consider is fully defined by the players, the time granularity at which the game is played, the actions the players can take, and the utility the players obtain as a result of the actions they take. We next discuss these different concepts in turn and illustrate them using a running example from an hospital.

Players: The game is played between the organization and its employees. We refer to the organization as $\mathcal{D}$ (*defender*). We subsume all employees into a single player $\mathcal{A}$ (*adversary*). In this paper, we are indeed considering a worst-case adversary, who would be able to control all employees and coerce them into adopting the strategy most damaging to the organization. In our running example, the players are the hospital and all the employees.

Round of play: In practice, audits are usually performed periodically. Thus, we adopt a discrete-time model for this game, where time points are associated with rounds. Each round of play corresponds to an audit cycle. We group together all of the adversary's actions in a given round.

Action space: $\mathcal{A}$ executes tasks, i.e., actions that are permitted as part of their job. We only consider tasks that can later be audited, e.g., through inspection of logs. We can distinguish $\mathcal{A}$'s tasks between legitimate tasks and violations of a specific privacy policy that the organization $\mathcal{D}$ must follow. Different types of violations may have a different impact on the organization. We assume that there are K different types of violations that $\mathcal{A}$ can commit (e.g., unauthorized access to a celebrity's records, unauthorized access to a family member's records, ...). We further assume that the severity of violations, in terms of economic impact on the organization, varies with the types.

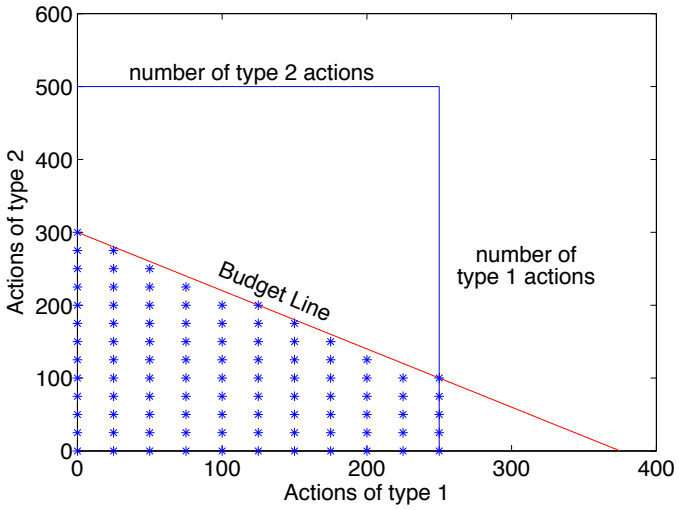

Figure 3.1: Feasible audit space, represented by the shaded area.

In each audit cycle, the adversary $\mathcal{A}$ chooses two quantities for each type k: the number of tasks she performs, and the number of such tasks that are violations. If we denote by U_k the maximum number of type k tasks that $\mathcal{A}$ can perform, then $\mathcal{A}$'s entire action space is given by $A \times A$ with $A = \prod_{i=1}^{K}\{1, \ldots, U_i\}$. In a given audit cycle, an action by $\mathcal{A}$ in the game is given by $\langle \vec{a}, \vec{v} \rangle$, where the components of $\vec{a}$ are the number of tasks of each type $\mathcal{A}$ performs, and the components of $\vec{v}$ are the number of violations of each type. Since violations are a subset of all tasks, we always have $\vec{v} \leq \vec{a}$. In our hospital example, we consider two types of patient medical records: access to celebrity records and access to regular person's record. A typical action may be 250 accesses to celebrity records with 10 of them being violations and 500 accesses to non-celebrity records with 50 of them being violations. Then $\mathcal{A}$'s action is $\langle \langle 250, 500 \rangle, \langle 10, 50 \rangle \rangle$.

We assume that the defender $\mathcal{D}$ can classify each adversary's task by types. However, $\mathcal{D}$ cannot determine whether a particular task is legitimate or a violation without investigating. $\mathcal{D}$ can choose to *inspect* or *ignore* each of $\mathcal{A}$'s tasks. We assume that inspection is perfect, i.e., if a violation is inspected then it is detected. The number of inspections that $\mathcal{D}$ can conduct is bounded by the number of tasks that $\mathcal{A}$ perform, and thus, $\mathcal{D}$'s actions are defined by a vector $\vec{s} \in A$, with $\vec{s} \leq \vec{a}$. That is, $\mathcal{D}$ chooses a certain number of tasks of each type to be inspected. Further, in each round t, $\mathcal{D}$ has a fixed budget B^t to allot to all inspections. We represent the

(assumed fixed) cost of inspection for each type of violation by $\vec{C}$. The budgetary constraints on $\mathcal{D}$ are thus given by $\vec{C} \cdot \vec{s} \leq B^t$ for all t. Continuing our hospital example, the maximum number of tasks of each type that $\mathcal{D}$ can inspect is $\langle 250, 500 \rangle$. Assuming a budget of 1500 and cost of inspection vector $\langle 4, 5 \rangle$, $\mathcal{D}$'s inspection space is further constrained, and then the feasible inspections are $\{\langle x, y \rangle \mid 4x + 5y \leq 1500, 0 \leq x \leq 250, 0 \leq y \leq 500\}$. The discrete feasible audit points are indicated (not all points are shown) with the asterisks in Figure 6.1.

Violation detection: Given the budgetary constraints $\mathcal{D}$ faces, $\mathcal{D}$ cannot, in general, inspect all of $\mathcal{A}$'s tasks (i.e., $\vec{C} \cdot \vec{a} > B^t$). Hence, some violations may go undetected internally, but could be detected externally. Governmental audits, whistle-blowing, information leaks are all but examples of situations that could lead to external detection of otherwise unnoticed violations. We assume that there is a fixed exogenous probability p $(0 < p < 1)$ of an internally undetected violation getting caught externally.

Formally, we define the outcome of a single audit cycle as the outcome of the internal audit and the number of violations detected externally. Due to the probabilistic nature of all quantities, this outcome is a random variable. Let $\vec{\mathbf{O}}^t$ be the outcome for the t^{th} round. Then $\vec{\mathbf{O}}^t$ is a tuple $\langle \vec{\mathbf{O}}^t_{int}, \vec{\mathbf{O}}^t_{ext} \rangle$ of violations caught internally and externally. By our definitions, the probability mass function for $\vec{\mathbf{O}}^t_{int}$ is parameterized by $\langle \vec{a}, \vec{v} \rangle$ and $\vec{s}$, and the probability mass function for $\vec{\mathbf{O}}^t_{ext}$ conditioned on $\vec{\mathbf{O}}^t_{int}$ is parameterized by p. We make no assumptions about this probability mass function. Observe that, because not all tasks can be inspected, the organization does not get to know the exact number of violations committed by the employees, which makes this a game of imperfect information. In our hospital example, given that $\mathcal{A}$'s action is $\langle \langle 250, 500 \rangle, \langle 10, 50 \rangle \rangle$. In one possible scenario the hospital performs $\langle 125, 200 \rangle$ inspections. These inspections result in $\langle 7, 30 \rangle$ violations detected internally and $\langle 2, 10 \rangle$ violations detected externally.

Utility function: Since we consider a worst-case adversary, $\mathcal{A}$'s payoff function is irrelevant to our model. On the other hand, the utility function of $\mathcal{D}$ influences the organization's strategy. We define $\mathcal{D}$'s utility as the sum of $\mathcal{D}$'s *reputation* and the cost of inspecting $\mathcal{A}$'s actions. In essence, $\mathcal{D}$ has to find the right trade-off between inspecting frequently (which incurs high costs) and letting violations occur (which degrades its reputation, and thus translates to lost revenue).

We assume that the cost of inspection is linear in the number of inspections for each type of action. Hence, if $\mathcal{D}$'s action is $\vec{s}$ then inspection costs are $\vec{C} \cdot \vec{s}$. In our running example of the hospital, this cost is $\langle 4, 5\rangle.\langle 125, 200\rangle = 1500$, which is the also the full budget in our example.

We assume that any violation caught (internally, or externally) in a round affects $\mathcal{D}$'s reputation not only in that round, but also in future rounds and that the exact effect in any future round is known. We capture this by defining a function $r_k^t : \{1, \ldots, U_k\} \times \{1, \ldots, U_k\} \times \mathbb{N} \to \mathbb{R}$ for each type k of violation. In round t, r_k^t takes as inputs the number of violations of type k detected internally, the number of violations of type k caught externally, and an integer argument τ. r_k^t outputs the effect of the violations (measured as the loss in reputation) occurring in round t on $\mathcal{D}$'s reputation in round $t + \tau$. We assume that violations of a given type always have the same effect on reputation, that is, r_k^t is actually independent of t, which allows us to use the shorthand notation r_k from here on.

Violations caught far in the past should have a lesser impact on reputation than recently caught violations, thus, r_k should be monotonically decreasing in the argument τ. We further assume violations are forgotten after a finite amount of rounds m, and hence do not affect reputation further. In other words, if $\tau \geq m$ then for any type k, any round t, and any tuple of violations caught $\langle \vec{O}_{int}^t[k], \vec{O}_{ext}^t[k]\rangle$, $r_k(\vec{O}_{int}^t[k], \vec{O}_{ext}^t[k], \tau) = 0$.

Moreover, externally caught violations should have a worse impact on reputation than internally detected violations, otherwise the organization has a trivial incentive never to inspect. Formally, r_k has the following property. If for any two realized outcomes $\vec{O}^l$ and $\vec{O}^j$ at rounds l and j, we have $\vec{O}_{int}^l[k] + \vec{O}_{ext}^l[k] = \vec{O}_{int}^j[k] + \vec{O}_{ext}^j[k]$ (i.e., same number of total violations of type k in rounds j and l) and $\vec{O}_{ext}^l[k] > \vec{O}_{ext}^j[k]$ (i.e., for type k, the number of violations caught externally is more than the number caught internally) then for any τ such that $0 \leq \tau < m$, $r_k(\vec{O}^l[k], \tau) > r_k(\vec{O}^j[k], \tau)$.

We use r_k to define a measure of reputation. Because, by construction, violations only affect at most m rounds of play, we can write the reputation $\mathbf{R}_0$ of the organization at round t as a random variable function of the probabilistic outcomes

$\vec{\mathbf{O}}^t, ..., \vec{\mathbf{O}}^{t-m+1}$:

$$\mathbf{R}_0(\vec{\mathbf{O}}^t, ..., \vec{\mathbf{O}}^{t-m+1}) = R - \sum_{k=1}^{K} \sum_{j=t-m+1}^{t} r_k(\vec{\mathbf{O}}^j[k], t-j) \ ,$$

where R is the maximum possible reputation. We assume that at the start of the game the reputation is R, and that r_k's are so that $\mathbf{R}_0$ is always non-negative.

We cannot, however, directly use $\mathbf{R}_0$ in our utility function. Indeed, $\mathbf{R}_0$ is history-dependent, and the repeated game formalism requires that the utility function be independent of past history. Fortunately, a simple construction allows to closely approximate the actual reputation, while at the same time removing dependency on past events. Consider the following function $\mathbf{R}$:

$$\mathbf{R}(\vec{\mathbf{O}}^t) = R - \sum_{k=1}^{K} \sum_{j=0}^{m-1} r_k(\vec{\mathbf{O}}^t[k], j) \ .$$

Rather than viewing reputation as a function of violations that occurred in the past, in round t, the reputation function $\mathbf{R}$ instead immediately accounts for reputation losses that will be incurred in the future (in rounds $t+\tau$, $0 \leq \tau < m$) due to violations occurring in round t.

While $\mathbf{R}$ and $\mathbf{R}_0$ are different reputation functions, when we compute the difference of their averages over T rounds, denoting by $\vec{v}_{\max}$ the maximum possible number of violations, we obtain:

$$\frac{1}{T} \sum_{\tau=t}^{t+T} |\mathbf{R}(\vec{\mathbf{O}}^\tau) - \mathbf{R}_0(\vec{\mathbf{O}}^\tau, \ldots, \vec{\mathbf{O}}^{\tau-m})| \leq \frac{1}{T} \sum_{k=1}^{K} \sum_{j=1}^{m-1} j \cdot r_k(\vec{v}_{max}(k), j) \ .$$

The right-hand side of the above inequality goes to zero as T grows large. Hence, using $\mathbf{R}$ to model reputation instead of $\mathbf{R}_0$ does not significantly impact the utility function of the defender. We define the utility function at round t in the repeated game by the random variable

$$\mathbf{L}^t(\langle \vec{a}^t, \vec{v}^t \rangle, \vec{s}^t) = \mathbf{R}(\vec{\mathbf{O}}^t) - \vec{C} \cdot \vec{s}^t \ .$$

Since utility gains are only realized through loss reduction, we will equivalently refer to $\mathbf{L}$ as a loss function from here on.

An example of the loss of reputation function r_k is $r_k(O,t) = c_k(O_{int}+2\times O_{ext})\delta^t$ for $0 \leq t < m$ and $r_k(O,t) = 0$ for $t \geq m$, where $\delta \in (0,1)$. Observe that r_k decreases with t and puts more weight on external violations. Also for the same number of violations and same value for argument t r_k has different values for different types of violations due to c_k that varies with the types. Then, considering only one type of violation, the loss function can be written as

$$\mathbf{L}^t(\langle \vec{a}^t, \vec{v}^t \rangle, \vec{s}^t) = R - c_1 \sum_{j=0}^{m-1} (\mathbf{O}^t_{int} + 2 \times \mathbf{O}^t_{ext})\delta^j - \vec{C} \cdot \vec{s}^t \ .$$

Observe that we can expand the summation in the above equation to get $c_1(1 + \delta ... + \delta^{m-1})\mathbf{O}^t_{int} + 2c_1(1 + \delta ... + \delta^{m-1})\mathbf{O}^t_{ext}$. Then let $R_{int} = c_1(1 + \delta ... + \delta^{m-1})$ and similarly let $R_{ext} = 2c_1(1 + \delta ... + \delta^{m-1})$. We can rewrite the loss equation above as

$$\mathbf{L}^t(\langle \vec{a}^t, \vec{v}^t \rangle, \vec{s}^t) = R - R_{int} \cdot \mathbf{O}^t_{int} - R_{ext} \cdot \mathbf{O}^t_{ext} - \vec{C} \cdot \vec{s}^t \ .$$

3.3 Audit Mechanism and Property

In this section, we present our audit mechanism RMA and the main theorem that characterizes its property. RMA prescribes the number of tasks of each type that the defender should inspect in each round of the repeated game. The property compares the loss incurred by the defender when she follows RMA to the loss of a hypothetical defender who has perfect knowledge of how many violations of each type occurred in each round, but must select one fixed action $\vec{s}$ to play in every round. In particular, we obtain exact bounds on the defender's regret and demonstrate that the average regret across all rounds converges to a small value relatively quickly.

3.3.1 Audit Mechanism

Our Regret Minimizing Audit (RMA) mechanism is presented as Algorithm 1. In each round of the game, RMA prescribes how many tasks of each type the defender should inspect. It does so by maintaining weights for each possible defender action (referred to as "experts" following standard terminology in the learning literature)

and picking an action with probability proportional to the weight of that action. For example, in a hospital audit, with two types of tasks—celebrity record access and regular record access—the possible defender actions $\vec{s}$ are of the form $\langle k_1, k_2 \rangle$ meaning that k_1 celebrity record accesses and k_2 regular record accesses are inspected. The weights are updated based on an estimated loss function, which is computed from the observed loss in each round. γ is a learning parameter for RMA. Its value is less than but close to 1. We show how to choose γ in sub-section 3.3.2.

Algorithm 1: RMA

Set $w_{\vec{s}}^0 = 1$ for each expert.

while *true* **do**

- $\langle \vec{a}^t, \vec{v}^t \rangle \leftarrow$ action of the adversary in round t.
- $\mathsf{AWAKE}^t \leftarrow \{\vec{s} : \vec{s} \leq \vec{a}^t \wedge \vec{C} \cdot \vec{s} \leq B^t \}$
- $W^t = \sum_{\vec{s} \in \mathsf{AWAKE}^t} w_{\vec{s}}^t$
- $p_{\vec{s}}^t \leftarrow \frac{w_{\vec{s}}^t}{W^t}$ for $\vec{s} \in \mathsf{AWAKE}$. Otherwise $p_{\vec{s}}^t \leftarrow 0$.
- Play $\vec{s}$ with probability $p_{\vec{s}}$ (randomly select one expert to follow).
- $\tilde{\mathbf{L}} \leftarrow \mathsf{est}\left(\vec{\mathbf{O}}^t, \vec{s}^t\right)$.
- $\tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA}) = \sum_{\vec{s}} p_{\vec{s}}^t \tilde{\mathbf{L}}^t(\vec{s})$
- **for** $\vec{s} \in \mathsf{AWAKE}^t$ **do**
  - $w_{\vec{s}}^{t+1} \leftarrow w_{\vec{s}}^t \gamma^{\tilde{\mathbf{L}}^t(\vec{s}) - \gamma \tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA})}$

RMA is fast and could be run in practice. Specifically, the running time of RMA is $O(N)$ per round where N is the number of experts.

In more detail, RMA maintains weights $w_{\vec{s}}^t$ for all experts [16]. $w_{\vec{s}}^t$ is the weight of the expert before round t has been played. Initially, all experts are equally weighted. In each round, an action is probabilistically selected for the defender. As discussed in the Section 3.2 there are two factors that constrain the set of actions available to the defender: the number of tasks performed by the adversary and the budget available for audits. In our hospital example we had the feasible audit space as $\{\langle x, y \rangle \mid 4x + 5y \leq 1500, 0 \leq x \leq 250, 0 \leq y \leq 500\}$. These considerations motivate the definition of the set AWAKE^t of experts that are awake in round t (Step 1). Next, from this set of awake experts, one is chosen with probability $p_{\vec{s}}^t$ proportional to the weight of that expert (Steps 2, 3, 4). Continuing our hospital example, 250 celebrity

record accesses and 500 regular record accesses will be inspected with probability 0.3 in a round if the expert $\langle 250, 500\rangle$ is awake in that round and its weight divided by the total weight of all the experts who are awake is 0.3. Technically, this setting is close to the setting of sleeping experts in the regret minimization literature [15, 17].

However, we also have to deal with imperfect information. Since only one action (say $\vec{s}^t$) is actually played by the defender in a round, she observes an outcome $\vec{\mathbf{O}}^t$ for that action. For example, the $\langle 250, 500\rangle$ inspection might have identified 5 celebrity record access violations and 3 regular record access violations internally; the same number of violations may have been detected externally. Based on this observation, RMA uses an algorithm est to compute an estimated loss function $\tilde{\mathbf{L}}$ for *all* experts (not just the one she played). We describe properties of the loss function for which this estimation is accurate in subsection 3.5. We also provide an example of a natural loss function that satisfies these properties. Finally, the estimated loss function is used to update the weights for all the experts who are awake. Intuitively, the multiplicative weight update ensures that the weights of experts who performed better than their current distribution increase and the weights for those who performed worse decrease. In RMA the weight for expert $\vec{s}$ increases when $\tilde{\mathbf{L}}^t(\vec{s}) - \gamma\tilde{\mathbf{L}}^t(\mathsf{RMA})$ is negative, i.e., $\tilde{\mathbf{L}}^t(\vec{s}) < \gamma\tilde{\mathbf{L}}^t(\mathsf{RMA})$, and since γ is close to 1, the loss of expert $\vec{s}$ is less than the loss of RMA, i.e., the expert $\vec{s}$ performed better than RMA.

3.3.2 Property

The RMA mechanism provides the guarantee that the defender's *regret* is minimal. *Regret* is a standard notion from the online learning literature. Intuitively, regret quantifies the difference between the loss incurred by the defender when she follows RMA and the loss of a hypothetical defender who has perfect knowledge of how many violations of each type occurred in each round, but must select one fixed action (or expert) to play in every round. Our main theorem establishes exact bounds on the defender's regret.

Let T denote the total number of rounds played, $I(t)$ be a *time selection function* whose output is either 0 or 1, $\mathbf{L}^t(\vec{s}) = \mathbf{L}^t\left(\langle \vec{a}^t, \vec{v}^t\rangle, \vec{s}\right)$ be the loss function at time t after the adversary has played $\langle \vec{a}^t, \vec{v}^t\rangle$, and $p^t_{\vec{s}}$ be the probability of choosing the

action $\vec{s}$ in round t while following RMA. We define the total loss of RMA as follows:

$$\textbf{Loss}\left(\mathsf{RMA}, I\right) = \sum_{t=1}^{T} \sum_{\vec{s}} I(t) p_{\vec{s}}^{t} \mathbf{L}^{t}(\vec{s}) \ .$$

Similarly for each fixed expert $\vec{s}$ we set the following loss function

$$\textbf{Loss}\left(\vec{s}, I\right) = \sum_{t=1}^{T} I(t) \mathbf{L}^{t}(\vec{s}) \ .$$

We use $\textbf{Regret}\left(\mathsf{RMA}, \vec{s}\right)$ to denote our regret in hindsight of not playing the fixed action $\vec{s}$ when it was available. Formally,

$$\textbf{Regret}\left(\mathsf{RMA}, \vec{s}\right) = \textbf{Loss}\left(\mathsf{RMA}, I_{\vec{s}}\right) - \textbf{Loss}\left(\vec{s}, I_{\vec{s}}\right).$$

Here, $I_{\vec{s}}(t)$ is the time selection function that selects only the times t that action $\vec{s}$ is available.

As before, we use N to denote the total number of fixed actions available to the defender. If T is known in advance we can obtain the bound in Theorem 1 below by setting γ to be $1 - \sqrt{\frac{2 \ln N}{T}}$. Otherwise, if T is not known in advance we can dynamically tune γ to obtain similar bounds. See Remark 7 for more details on dynamic tuning.

Theorem 1. *For all* $\epsilon > 0$,

$$\Pr\left[\ \exists \vec{s}, \quad \begin{array}{c} \frac{\textbf{Regret}(\mathsf{RMA}, \vec{s})}{T} \geq 2\sqrt{2\frac{\ln N}{T}} + \\ 2\sqrt{\frac{2\ln\left(\frac{4N}{\epsilon}\right)}{T}} + \frac{2}{T}\ln N \end{array} \ \right] \leq \epsilon \ .$$

Remark 1. *To prove this theorem we need to make several reasonable assumptions about the accuracy of our loss function estimator* est. *We discuss these assumptions in Section 3.5. We also assume that losses have been scaled so that* $\mathbf{L}^{t}(x) \in [0, 1]$.

Remark 2. *This bound is a worst case regret bound. The guarantee holds against* any *attacker.* **Regret** *may typically be lower than this, e.g. when hospital employees do not behave adversarially.*

In order to understand what this bound means, consider the following example scenario. Suppose that an employee at a hospital can access two types of medical

records—celebrity or regular. The defender can choose to inspect accesses of a certain type *lightly, moderately,* or *heavily.* In this case, the defender has $N = 9$ possible pairs of actions in each round. If the hospital performs daily audits (which some hospitals currently do for celebrity record accesses) over a 5 year period, then $T = 365 \times 5 = 1825$. For simplicity, assume that each action $\vec{s}$ is available every day. In this case, the theorem guarantees that except with probability $\epsilon = \frac{1}{100}$, the average regret of RMA does not exceed 29%:

$$\frac{\textbf{Regret}\left(\mathsf{RMA}, \vec{s}\right)}{T} < 0.29 \ .$$

Note that there are several existing algorithms for regret minimization in games with imperfect information [35, 56, 57, 58]. These algorithms do guarantee that as $T \to \infty$ the average regret will tend to 0, but the convergence rate is unacceptably slow for our audit model (see Section **??** for a more detailed comparison). The convergence rate of RMA is significantly faster. Also, in contrast to prior work, we focus on *exact* (not *asymptotic*) regret bounds for our algorithm. This is important because in practice we care about the value of the bound for a fixed value of T (as in the example above), not merely that it tends to 0 as $T \to \infty$.

3.4 Discussion

A few characteristics of the model and algorithms described above may not necessarily be evident from the technical presentation given above, and warrant further discussion.

Figure 3.2 shows the variation of average regret with time for different values of N and ϵ. As can be seen, the RMA algorithm produces smaller average regret bounds for higher values of time T and lower values of N. In other words, and quite intuitively, a high audit frequency ensures low regret. Some medical centers carry out audits every week; RMA is particularly appropriate for such high frequency audits. Lower values of N means that RMA's performance is compared to fewer fixed strategies and hence yields lower regret. One situation in which N could be low is when the fixed strategies correspond to discrete levels of audits coverage used by the organization. Also, higher values of ϵ yield smaller average regret bounds. Indeed, ϵ is a measure of uncertainty on the stated bound. Thus, when higher values

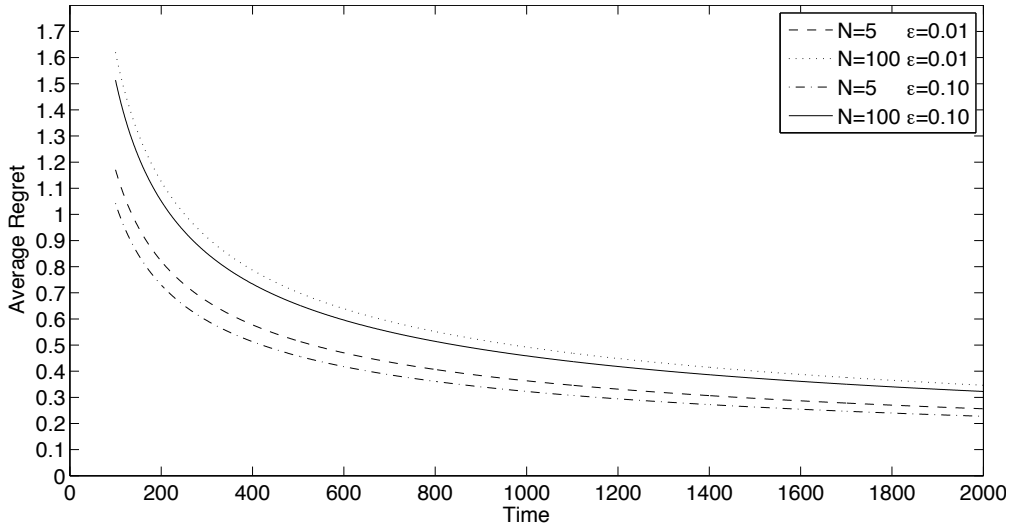

Figure 3.2: Worst case Average Regret vs Time for different values of N and ϵ

of ϵ are tolerable, we obtain tighter regret bounds, but at the expense of greater uncertainty on whether those bounds are met.

We have already noted that all actions may not be available at all times. The result in Theorem 1 bounds the average regret taken over all rounds of the game. It is easy to modify the proof of Theorem 1 to obtain average regret bounds for each expert such that the average is taken over the time for which that expert is awake. The bound thus obtained is of the same *order* as the bound in Theorem 1, but all instance of T in Theorem 1 are replaced by $T_{\vec{s}}$, where $T_{\vec{s}}$ is the time for which expert $\vec{s}$ is awake. Similar result for the traditional sleeping experts setting can be found in Blum et al. [17]. The modified bound equation exhibits the fact that the average regret bound for a given inspection vector (average taken over the time for which that inspection was available) depends on how often this inspection vector is available to the defender. If a given inspection vector is only available for a few audit cycles, the average regret bound may be relatively high. The situation is analogous to whitewashing attacks [59], where the adversary behaves in a compliant manner for many rounds to build up reputation, attacks only once, and immediately leaves the game after the attack. For instance, a spy infiltrates an organization, becomes a trusted member by behaving as expected, and then suddenly steals sensitive data. However, we argue that, rather than being an auditing issue, whitewashing attacks can be handled by a different class of mechanisms, e.g., that prevent the adversary

from vanishing once she has attacked.

Furthermore, RMA guarantees low *average* regret compared to playing a fixed action (i.e., inspection vector) in the audit cycle in which that action was available; it does not guarantee violations will not happen. In particular, if a certain type k of violation results in catastrophic losses for the organization (e.g., losses that threaten the viability of the organization itself), tasks of type k should always be fully inspected.

While the discussion surrounding the RMA algorithm focused only on one kind of experts (which recommend how many tasks of each type to inspect in any round), RMA applies more generally to any set of experts. For example, we could include an expert who recommends a low inspection probability when observed violations are below a certain threshold and a higher inspection probability when observed violations are above that threshold. Over time, the RMA algorithm will perform as well as any such expert. Note. however, that if we make the size (N) of the set of experts too large, the regret bounds from Theorem 1 will be worse. Thus, in any particular application, the RMA algorithm will be effective if appropriate experts are chosen without making the set of experts too large.

Finally, we note that non-compliance with external privacy regulations may not only cause a loss of reputation for the organization, but can also result in fines being levied against the organization. For instance, a hospital found in violation of HIPAA provisions [55] in the United States will likely face financial penalties in addition to damaging its reputation. We can readily extend our model to account for such cases, by forcing the defender (organization) to perform some minimum level of audit (inspections) to meet the requirements stipulated in the external regulations. For example, we can constrain the action space available to the defender by removing strategies such as "never inspect." As long as the budgetary constraints allow the organization to perform inspections in addition to the minimal level of audit required by law, the guarantees provided by RMA still hold. Indeed, Theorem 1 holds as long as there is at least one awake expert in each round.

3.5 Estimating Losses

RMA uses a function $\mathsf{est}\left(\vec{\mathbf{O}}^t, \vec{s}^t\right)$ to estimate the loss function $\tilde{\mathbf{L}}^t$.

In this section, we formally define two properties—*accuracy* and *independence*; the regret bound in Theorem 1 holds for any estimator function that satisfies these two properties. We also provide an example of a loss function estimator algorithm that provably satisfies these properties, thus demonstrating that such estimator functions can in fact be implemented. The use of an estimator function and the characterization of its properties is a novel contribution of this paper that allows us to achieve significantly better bounds than prior work in the regret minimization literature for repeated games of imperfect information (see Section **??** for a detailed comparison).

Estimator Properties The function $\tilde{\mathbf{L}}^t = \mathsf{est}\left(\vec{\mathbf{O}}^t, \vec{s}^t\right)$ should be efficiently computable for practical applications. Note that the loss estimation at time t depends on the outcome $\vec{\mathbf{O}}^t$ (violations of each type detected internally and externally) and the defender's action $\vec{s}^t$ at time t. Intuitively, the function outputs an estimate of the loss function by estimating the number of violations of each type based on the detected violations of that type and the probability of inspecting each action of that type following the defender's action.

For each defender action (expert) $\vec{s}$, we define the random variable

$$\mathbf{X}_{\vec{s}}^t = \tilde{\mathbf{L}}^t(\vec{s}) - \mathbf{L}^t(\vec{s}).$$

Intuitively, $\mathbf{X}_{\vec{s}}^t$ is a random variable representing our estimation error at time t after the actions $\langle \vec{v}^t, \vec{a}^t \rangle$ and $\vec{s}^t$ have been fixed by the adversary and the defender respectively.

Because we have assumed that our loss functions are scaled so that $\tilde{\mathbf{L}}^t(\vec{s}), \mathbf{L}^t(\vec{s}) \in [0, 1]$ we have $\mathbf{X}_{\vec{s}}^t \in [-1, 1]$. This property of $\mathbf{X}_{\vec{s}}^t$ is useful in bounding the regret as we discuss later.

Formally, we assume the following properties about est:

1. **Accuracy:** $E\left[\mathbf{X}_{\vec{s}}^j\right] = 0$ for $0 \leq j \leq T$.
2. **Independence:** $\forall \vec{s}$, $\mathbf{X}_{\vec{s}}^1, \ldots, \mathbf{X}_{\vec{s}}^T$ are all independent random variables.

Any estimation scheme est that satisfies both properties can be plugged into RMA yielding the regret bound in Theorem 1. Informally, *accuracy* captures the idea that the estimate is accurate in an expected sense while *independence* captures

the idea that the error in the estimate in each round is independent of the error in all other rounds. We motivate these properties by way of an example.

Remark 3. *In fact if our estimation scheme only satisfied δ-accuracy, i.e., $\left|E\left[\mathbf{X}_{\vec{s}}^{j}\right]\right| < \delta$, then we could still guarantee that the average regret bounds from Theorem 1 still hold with an extra additive term δ. Formally, the following property holds: for all $\epsilon \in (0,1)$*

$$\Pr\left[\begin{array}{cc} \exists \vec{s}, & \frac{\mathbf{Regret}(\mathsf{RMA},\vec{s})}{T} \geq \delta + 2\sqrt{2\frac{\ln N}{T}} + \\ & 2\sqrt{\frac{2\ln\left(\frac{4N}{\epsilon}\right)}{T}} + \frac{2}{T}\ln N \end{array}\right] \leq \epsilon \ .$$

Example Loss Function We return to our running example of the hospital. We use the example reputation (loss) function from the previous section:

$$\mathbf{L}^t(\vec{s}) = R - \left(\vec{\mathbf{O}}_{int}^t \cdot \vec{R}_{int} + \vec{\mathbf{O}}_{ext}^t \cdot \vec{R}_{ext} + \vec{C} \cdot \vec{s}\right) \ .$$

To simplify our presentation we assume that there is only one type of violation. It is easy to generalize the loss function that we present in this example to include multiple types of violations.

$$\mathbf{L}^t(s) = R - \left(\mathbf{O}_{int}^t \times R_{int} + \mathbf{O}_{ext}^t \times R_{ext} + C \times s\right) \ .$$

Here $\mathbf{O}_{int}^t$ represents the number of violations caught internally after the actions $\langle v^t, a^t\rangle$ and s^t are played by the adversary and the defender respectively, R_{int} (resp. R_{ext}) captures the damage to the hospital's reputation when a violation is caught internally (resp. externally), and C is the cost of performing one inspection. Notice that

$$E\left[\mathbf{O}_{ext}^t \times R_{ext}\right] = p\left(v^t - E\left[\mathbf{O}_{int}^t\right]\right) \times R_{ext} \ ,$$

where p is the probability that an undetected violation gets caught externally. Therefore,

$$E\left[\mathbf{L}^t(s)\right] \ = \ R - \left(E\left[\mathbf{O}_{int}^t\right](R_{int} - p \times R_{ext}) + p \times v^t \times R_{ext} + C \times s\right) \ .$$

We can set $R' = (R_{int} - p \times R_{ext})$ and then

$$E\left[\mathbf{L}^t(s)\right] = R - \left(E\left[\mathbf{O}^t_{int}\right] \times R' + p \times v^t \times R_{ext} + C \times s\right) \ .$$

In our loss model, we allow the defender to use any recommendation algorithm REC that sorts all a^t actions at time t and probabilistically recommends s^t actions to inspect. We let $p_d \leq 1$ denote the probability that the d^{th} inspection results in a detected violation, where this probability is over the coin flips of the recommendation algorithm REC. Because this probability is taken over the coin flips of REC the outcome $\mathbf{O}^t_{int}$ is independent of previous outcomes once $\langle \vec{a}^t, \vec{v}^t \rangle, \vec{s}^t$ have been fixed.

For example, a naive recommendation algorithm REC might just select a few actions uniformly at random and recommend that the defender inspect these actions. In this case $p_j = \frac{v^t}{a^t}$ for each j. (Remember that in this example we consider only one type of violation, so v^t and a^t are scalars). If REC is more clever, then we will have $p_1 > \frac{v^t}{a^t}$. In this case the p_j's will also satisfy diminishing returns ($p_j > p_{j+1}$).

We assume that inspection is perfect, i.e., if we inspect a violation it will be caught with probability 1. Thus, if we inspect all a^t actions we would catch all v^t violations, i.e.,

$$\sum_{j=1}^{a^t} p_j = v^t.$$

Set $p_j = v^t \left(\frac{1-\beta}{1-\beta^{a^t}}\right) \beta^{j-1}$, where the parameter β could be any value in $(0, 1)$. Notice that

$$\sum_{j=1}^{a} p_j = v^t \left(\frac{1-\beta}{1-\beta^{a^t}}\right) \sum_{j=0}^{a^t-1} \beta^j = v^t \ ,$$

and $p_j > p_{j+1}$ so our model does satisfy diminishing returns. Furthermore, if $\beta = \max\{1 - \frac{1}{a^t}, \frac{1}{2}\}$ then we have $p_j \leq 1$ for each j. We can express $E\left[\mathbf{O}_{int}\right] = \sum_{i=1}^{s^t} p_j$.

$$E\left[\mathbf{L}^t(s)\right] = R - \left(R' \sum_{i=1}^{s^t} p_j + p \times v^t \times R_{ext} + C \times s\right) \ .$$

Example Loss Estimator Our loss function estimator $\mathsf{est}\left(\mathbf{O}^t_{int}, s^t\right)$ is given in Algorithm 2.

Assuming that the defender understands the accuracy of his recommendation

Algorithm 2: Example of estimator $\mathsf{est}\left(\mathbf{O}_{int}^t, s^t\right)$

Data: $\mathbf{O}_{int}^t, s^t$

$\tilde{\mathbf{v}}^t \leftarrow \frac{1-\beta^{a^t}}{1-\beta^{s^t}}\mathbf{O}_{int}^t$

$$\tilde{\mathbf{L}}(x) \leftarrow R - \begin{pmatrix} R' \times \tilde{\mathbf{v}}^t \times \sum_{j=1}^{x}\left(\frac{1-\beta}{1-\beta^a}\beta^{j-1}\right) \\ +p \times \tilde{\mathbf{v}}^t \times R_{ext} + C \times x \end{pmatrix}$$

Result: $\tilde{\mathbf{L}}^t$

algorithm REC[1] β is a known quantity so that this computation is feasible and can be performed quickly. Independence of the random variables $\mathbf{X}_s^t$ follows from the independence of $\mathbf{O}_{int}^t$. Now we verify that our estimator satisfies our accuracy condition.

Claim 1. *When* $\tilde{\mathbf{L}}^t = \mathsf{est}\left(\mathbf{O}_{int}^t, s^t\right)$ *from Algorithm 2* $E\left[\mathbf{X}_s^t\right] = 0$.

The proof can be found in Appendix A.1. The main insight here is that because of the way the probabilities p_j's are scaled, the expectation of the estimated number of violations is equal to the number of actual violations, i.e. $E\left[\tilde{\mathbf{v}}^t\right] = v^t$. Consequently, the expected value of the error turns out to be 0, thus satisfying the accuracy property.

Remark 4. *If there are multiple types of actions then we can estimate* $\tilde{\mathbf{v}}_{\mathbf{k}}^t$*, the number of violations of type* k *at time* t*, separately for each* k*. To do this* est *should substitute* $\vec{\mathbf{O}}_{int}^t[k]$ *and* $\vec{s}^t[k]$ *for* $\mathbf{O}_{int}^t$ *and* v^t *and set*

$$\tilde{\mathbf{v}}_{\mathbf{k}}^t := \left(\frac{1-\beta^{a^t}}{1-\beta^{s^t}}\right)\mathbf{O}_{int}^t \ .$$

Then we have

$$\tilde{\mathbf{L}}^t(x) = R - \begin{pmatrix} \vec{R}' \cdot \langle\tilde{\mathbf{v}}_{\mathbf{k}}^t\rangle_k \times \frac{1-\beta^{x[k]}}{1-\beta^a} \\ +p \times \langle\tilde{\mathbf{v}}_{\mathbf{k}}^t\rangle_k \cdot \vec{R}_{ext} + \vec{C} \cdot \vec{x} \end{pmatrix} \ .$$

3.6 Proof Outline

In this section, we present the outline of the proof of our main theorem (Theorem 1) which establishes high probability regret bounds for RMA. The complete proof

[1]If the algorithm recommends actions uniformly at random, then the accuracy is certainly understood.

is in the appendices. The proof proceeds in two steps:

1. We first prove that RMA achieves low regret with respect to the estimated loss function using standard results from the literature on regret minimization [15, 17].
2. We then prove that with high probability the difference between regret with respect to the actual loss function and regret with respect to the estimated loss function is small. This step makes use of the two properties—accuracy and independence—of the estimated loss function est presented in the previous section and is the novel part of the proof. The key technique used in this step are the Hoeffding inequalities [60].

Let T denote the total number of rounds played, $T_{\vec{s}}$ denote the total number of rounds that the action $\vec{s}$ was awake and I as before is a time selector function. We define

$$\widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, I\right) = \sum_{t=1}^{T} \sum_{\vec{s}} I(t) p_{\vec{s}}^{t} \tilde{\mathbf{L}}^{t}(\vec{s}) \ ,$$

to be our total estimated loss. Notice that $\widetilde{\mathbf{Loss}}$ is the same as $\mathbf{Loss}$ except that we replaced the actual loss function $\mathbf{L}^t$ with the estimated loss function $\tilde{\mathbf{L}}^t$. We define $\widetilde{\mathbf{Loss}}\left(\vec{s}, I\right)$ and $\widetilde{\mathbf{Regret}}\left(\mathsf{RMA}, \vec{s}\right)$ in a similar manner by using the estimated loss function.

Lemma 1 and Lemma 2 below bound the regret with respect to the estimated loss function. They are based on standard results from the literature on regret minimization [15, 17]. We provide full proofs of these results in Appendix A.3.

Lemma 1. *For each expert $\vec{s}$ we have*

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq \frac{1}{L-1} T + 2L \ln N \ ,$$

where N is the total number of experts and γ, our learning parameter has been set to $\gamma = 1 - \frac{1}{L}$

Remark 5. *We would like to bound our average regret:*

$$\frac{\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s})}{T_{\vec{s}}} \ .$$

There is a trade-off here in the choice of L. If L is too large, then $L \ln N$ will be

large. If L is too small, then $\frac{1}{L-1}T$ will be large. If we know T in advance, then we can tune our learning parameter γ to obtain the best bound by setting

$$L = \sqrt{\frac{T}{2\ln N}} + 1 \ .$$

After substituting this value for L in Lemma 1, we immediately obtain the following result:

Lemma 2. *For each expert $\vec{s}$ we have*

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \ \leq \ 2\sqrt{2T\ln N} + 2\ln N \ ,$$

where N is the total number of experts and learning parameter $\gamma = 1 - \sqrt{\frac{2\ln N}{T}}$.

Remark 6. *This shows that* RMA *can achieve low regret with respect to the estimated loss functions $\tilde{\mathbf{L}}^t$. This completes step 1 of the proof. We now move on to step 2.*

Notice that we can write our actual loss function ($\mathbf{Loss}\,(\mathsf{RMA}, I)$) in terms of our estimated loss function $\left(\widetilde{\mathbf{Loss}}\,(\mathsf{RMA}, I)\right)$ and $\mathbf{X}^t_{\vec{s}}$.

Fact 1.

$$\mathbf{Loss}\,(\mathsf{RMA}, I) = \widetilde{\mathbf{Loss}}\,(\mathsf{RMA}, I) + \sum_{t=1}^{T} I(t)\mathbf{X}^t_{\vec{s}} \ .$$

We know that $E\left[\mathbf{X}^t_{\vec{s}}\right] = 0$ (from the accuracy property of the loss estimation function) and that $\mathbf{X}^1_{\vec{s}}, \ldots \mathbf{X}^T_{\vec{s}}$ are independent (from the independence property), so we can apply the Hoeffding inequalities to bound $\sum_t \mathbf{X}^t_{\vec{s}}$ obtaining Lemma 3. Appendix A.2 contains a description of the inequalities and the full proof of the following lemma.

Lemma 3.

$$\Pr\left[\exists \vec{s}, \mathbf{Regret}\,(\mathsf{RMA}, \vec{s}) - \widetilde{\mathbf{Regret}}\,(\mathsf{RMA}, \vec{s}) \geq 2K\right] \leq \epsilon \ ,$$

where $K = \sqrt{2T\ln\left(\frac{4N}{\epsilon}\right)}$.

After straightforward algebraic substitution we can obtain our main result in Theorem 1 by combining Lemma 3 with Lemma 2 (see Appendix A.4).

Observe that the optimal value of γ is dependent on T. But, it is conceivable that the time T for which the game is played is not known in advance. The following

remark shows that we can overcome this problem by choosing a dynamic value for γ. This makes RMA usable in the real world.

Remark 7. *Even if we don't know T in advance we can tune γ dynamically using a technique from [61]. We set*

$$\gamma_t = \frac{1}{1-\alpha_t} \ ,$$

where

$$\alpha_t = \sqrt{2\frac{\ln N}{L^t - 1}} \ .$$

where L^t is the minimum loss of any expert till time t (calculated using the estimated loss). Before playing round t we recompute the weights $w_{\vec{s}}^t$, pretending that our learning parameter γ had been set to γ_t from the beginning i.e.

$$w_{\vec{s}}^t = \gamma_t^{\sum_{i=1}^{t} I_{\vec{s}}(t)\left(\tilde{\mathbf{L}}^t(\vec{s}) - \gamma_t \tilde{\mathbf{L}}^t(\mathsf{RMA})\right)} \ .$$

In this case our final guarantee (similar to Theorem 1) would be that:

$$\Pr\left[\exists \vec{s}, \begin{array}{c} \frac{\mathbf{Regret}(\mathsf{RMA},\vec{s})}{T} \geq 2\sqrt{2\frac{\ln N}{T}} + \\ 2\sqrt{\frac{2\ln\left(\frac{4N}{\epsilon}\right)}{T}} + \frac{10}{T}\ln N + \\ \frac{4}{T}\left(\ln N\right)\left(\ln(1+T)\right) \end{array}\right] \leq \epsilon \ .$$

Chapter 4

Audit Game: Single Defender Resource

In a seminal paper, eminent economist Gary Becker [62] presented a compelling economic treatment of crime and punishment. He demonstrated that effective law enforcement involves optimal resource allocation to prevent and detect violations, coupled with appropriate *punishments* for offenders. He described how to optimize resource allocation by balancing the societal cost of crime and the cost incurred by prevention, detection and punishment schemes. While Becker focused on crime and punishment in society, similar economic considerations guide enforcement of a wide range of policies. In this and the following chapter, we present effective enforcement mechanisms for this broader set of policies in a rational adversary setting. Our study differs from Becker's in two significant ways—our model accounts for *strategic interaction* between the enforcer (or defender) and the adversary; and we design efficient algorithms for *computing* the optimal resource allocation for prevention or detection measures as well as punishments.

Our Model

We model the audit process as a game between a defender (e.g, a hospital) and an adversary (e.g., an employee). The defender audits a given set of targets (e.g., health record accesses) and the adversary chooses a target to attack. The defender's action space in the audit game includes two components. First, the allocation of its inspection resources to targets; this component also exists in a standard model of

security games [21]. Second, we introduce a continuous punishment rate parameter that the defender employs to deter the adversary from committing violations. However, punishments are not free and the defender incurs a cost for choosing a high punishment level. For instance, a negative work environment in a hospital with high fines for violations can lead to a loss of productivity (see [62] for a similar account of the cost of punishment). The adversary's utility includes the benefit from committing violations and the loss from being punished if caught by the defender. Our model is parametric in the utility functions. Thus, depending on the application, we can instantiate the model to either allocate resources for detecting violations or preventing them. This generality implies that our model can be used to study all the applications previously described in the security games literature [21].

To analyze the audit game, we use the Stackelberg equilibrium solution concept [63] in which the defender commits to a strategy, and the adversary plays an optimal response to that strategy. This concept captures situations in which the adversary learns the defender's audit strategy through surveillance or the defender publishes its audit algorithm. In addition to yielding a better payoff for the defender than any Nash equilibrium, the Stackelberg equilibrium makes the choice for the adversary simple, which leads to a more predictable outcome of the game. Furthermore, this equilibrium concept respects the computer security principle of avoiding "security through obscurity"— audit mechanisms like cryptographic algorithms should provide security despite being publicly known.

Our Results

Our approach to computing the Stackelberg equilibrium is based on the multiple LPs technique of Conitzer and Sandholm [24]. However, due to the effect of the punishment rate on the adversary's utility, the optimization problem in audit games has quadratic and non-convex constraints. The non-convexity does not allow us to use any convex optimization methods, and in general polynomial time solutions for a broad class of non-convex optimization problems are not known [**?**].

However, we demonstrate that we can efficiently obtain an additive approximation to our problem. Specifically, we present an additive fully polynomial time approximation scheme (FPTAS) to solve the audit game optimization problem. Our algorithm provides a K-bit precise output in time polynomial in K. Also, if the so-

lution is rational, our algorithm provides an exact solution in polynomial time. In general, the exact solution may be irrational and may not be representable in a finite amount of time.

4.1 The Audit Game Model

The audit game features two players: the defender (D), and the adversary (A). The defender wants to audit n targets $t_1, \ldots, t_n$, but has limited resources which allow for auditing only one of the n targets. Thus, a pure action of the defender is to choose which target to audit. A randomized strategy is a vector of probabilities $p_1, \ldots, p_n$ of each target being audited. The adversary attacks one target such that given the defender's strategy the adversary's choice of violation is the best response.

As our model is parametric in the utility functions, we do not use the explicit formulas for utility from the last section; this also helps in ease of exposition. Let the utility of the defender be $U_D^a(t_i)$ when audited target t_i was found to be attacked, and $U_D^u(t_i)$ when unaudited target t_i was found to be attacked. The attacks (violation) on unaudited targets are discovered by an external source (e.g. government, investigative journalists,...). Similarly, define the utility of the attacker as $U_A^a(t_i)$ when the attacked target t_i is audited, and $U_A^u(t_i)$ when attacked target t_i is not audited, excluding any punishment imposed by the defender. Attacks discovered externally are costly for the defender, thus, $U_D^a(t_i) > U_D^u(t_i)$. Similarly, attacks not discovered by internal audits are more beneficial to the attacker, and $U_A^u(t_i) > U_A^a(t_i)$.

The model presented so far is identical to security games with singleton and homogeneous schedules, and a single resource [31]. The additional component in audit games is punishment. The defender chooses a punishment "rate" $x \in [0, 1]$ such that if auditing detects an attack, the attacker is fined an amount x. However, punishment is not free—the defender incurs a cost for punishing, e.g., for creating a fearful environment. For ease of exposition, we model this cost as a linear function ax, where $a > 0$; however, our results directly extend to any cost function polynomial in x. Assuming $x \in [0, 1]$ is also without loss of generality as utilities can be scaled to be comparable to x. We do assume the punishment rate is fixed and deterministic; this is only natural as it must correspond to a consistent policy.

We can now define the full utility functions. Given probabilities $p_1, \ldots, p_n$ of

each target being audited, the utility of the defender when target t_* is attacked is

$$p_* U_D^a(t_*) + (1 - p_*) U_D^u(t_*) - ax.$$

The defender pays a fixed cost ax regardless of the outcome. In the same scenario, the utility of the attacker when target t_* is attacked is

$$p_*(U_A^a(t_*) - x) + (1 - p_*) U_A^u(t_*).$$

The attacker suffers the punishment x only when attacking an audited target.

Equilibrium. The Stackelberg equilibrium solution involves a commitment by the defender to a strategy (with a possibly randomized allocation of the resource), followed by the best response of the adversary. The mathematical problem involves solving multiple optimization problems, one each for the case when attacking t_* is in fact the best response of the adversary. Thus, assuming t_* is the best response of the adversary, the $*^{th}$ optimization problem P_* in audit games is

$$\begin{array}{ll} \max\limits_{p_i, x} & p_* U_D^a(t_*) + (1 - p_*) U_D^u(t_*) - ax \ , \\ \text{subject to} & \forall i \neq *.\ p_i(U_A^a(t_i) - x) + (1 - p_i) U_A^u(t_i) \\ & \qquad \leq p_*(U_A^a(t_*) - x) + (1 - p_*) U_A^u(t_*) \ , \\ & \forall i.\ 0 \leq p_i \leq 1 \ , \ \textstyle\sum_i p_i = 1 \ , \\ & 0 \leq x \leq 1 \ . \end{array}$$

The first constraint verifies that attacking t_* is indeed a best response. The auditor then solves the n problems $P_1, \ldots, P_n$ (which correspond to the cases where the best response is $t_1, \ldots, t_n$, respectively), and chooses the best solution among all these solutions to obtain the final strategy to be used for auditing. This is a generalization of the multiple LPs approach of Conitzer and Sandholm [24].

Inputs. The inputs to the above problem are specified in K bit precision. Thus, the total length of all inputs is $O(nK)$. Also, the model can be made more flexible by including a dummy target for which all associated costs are zero (including punishment); such a target models the possibility that the adversary does not attack any target (no violation). Our result stays the same with such a dummy target, but, an additional edge case needs to be handled—we discuss this case in a remark

at the end of Section 4.2.2.

4.2 Computing an Audit Strategy

Because the indices of the set of targets can be arbitrarily permuted, without loss of generality we focus on one optimization problem P_n ($* = n$) from the multiple optimization problems presented in Section 4.1. The problem has quadratic and non-convex constraints. The non-convexity can be readily checked by writing the constraints in matrix form, with a symmetric matrix for the quadratic terms; this quadratic-term matrix is indefinite.

However, for a fixed x, the induced problem is a linear programming problem. It is therefore tempting to attempt a binary search over values of x. This naïve approach does not work, because the solution may not be single-peaked in the values of x, and hence choosing the right starting point for the binary search is a difficult problem. Another naïve approach is to discretize the interval $[0, 1]$ into steps of ϵ', solve the resultant LP for the $1/\epsilon'$ many discrete values of x, and then choose the best solution. As an LP can be solved in polynomial time, the running time of this approach is polynomial in $1/\epsilon'$, but the approximation factor is at least $a\epsilon'$ (due to the ax in the objective). Since a can be as large as 2^K, getting an ϵ-approximation requires ϵ' to be $2^{-K}\epsilon$, which makes the running time exponential in K. Thus, this scheme cannot yield an FPTAS.

4.2.1 High-Level Overview

Fortunately, the problem P_n has another property that allows for efficient methods. Let us rewrite P_n in a more compact form. Let $\Delta_{D,i} = U_D^a(t_i) - U_D^u(t_i)$, $\Delta_i = U_A^u(t_i) - U_A^a(t_i)$ and $\delta_{i,j} = U_A^u(t_i) - U_A^u(t_j)$. $\Delta_{D,i}$ and Δ_i are always positive, and P_n reduces to:

$$
\begin{array}{ll}
\max\limits_{p_i, x} & p_n\Delta_{D,n} + U_D^u(t_n) - ax \ , \\
\text{subject to} & \forall i \neq n.\ p_i(-x - \Delta_i) + p_n(x + \Delta_n) + \delta_{i,n} \leq 0 \ , \\
& \forall i.\ 0 \leq p_i \leq 1 \ , \\
& \sum_i p_i = 1 \ , \\
& 0 \leq x \leq 1 \ .
\end{array}
$$

The main observation that allows for polynomial time approximation is that, at the optimal solution point, the quadratic constraints can be partitioned into a) those that are tight, and b) those in which the probability variables p_i are zero (Lemma 4). Each quadratic constraint corresponding to p_i can be characterized by the curve $p_n(x+\Delta_n)+\delta_{i,n} = 0$. The quadratic constraints are thus parallel hyperbolic curves on the (p_n, x) plane; see Figure 4.1 for an illustration. The optimal values p_n^o, x^o partition the constraints (equivalently, the curves): the constraints lying below the optimal value are tight, and in the constraints above the optimal value the probability variable p_i is zero (Lemma 5). The partitioning allows a linear number of iterations in the search for the solution, with each iteration assuming that the optimal solution lies between adjacent curves and then solving the sub-problem with equality quadratic constraints.

Next, we reduce the problem with equality quadratic constraints to a problem with two variables, exploiting the nature of the constraints themselves, along with the fact that the objective has only two variables. The two-variable problem can be further reduced to a single-variable objective using an equality constraint, and elementary calculus then reduces the problem to finding the roots of a polynomial. Finally, we use known results to find approximate values of irrational roots.

4.2.2 Algorithm and Main Result

The main result of our paper is the following theorem:

Theorem 2. *Problem P_n can be approximated to an additive ϵ in time $O(n^5K + n^4\log(\frac{1}{\epsilon}))$ using the splitting circle method [25] for approximating roots.*

Remark 8. *The technique of Lenstra et al. [64] can be used to exactly compute rational roots. Employing it in conjunction with the splitting circle method yields a time bound $O(\max\{n^{13}K^3,\ n^5K + n^4\log(1/\epsilon)\})$. Also, this technique finds an exact optimal solution if the solution is rational.*

Before presenting our algorithm we state two results about the optimization problem P_n that motivate the algorithm and are also used in the correctness analysis. The proof of the first lemma is in Appendix B.5.

Lemma 4. *Let p_n^o, x^o be the optimal solution. Assume $x^o > 0$ and $p_n^o < 1$. Then, at p_n^o, x^o, for all $i \neq n$, either $p_i = 0$ or $p_n^o(x^o + \Delta_n) + \delta_{i,n} = p_i(x^o + \Delta_i)$, i.e., the i^{th} quadratic constraint is tight.*

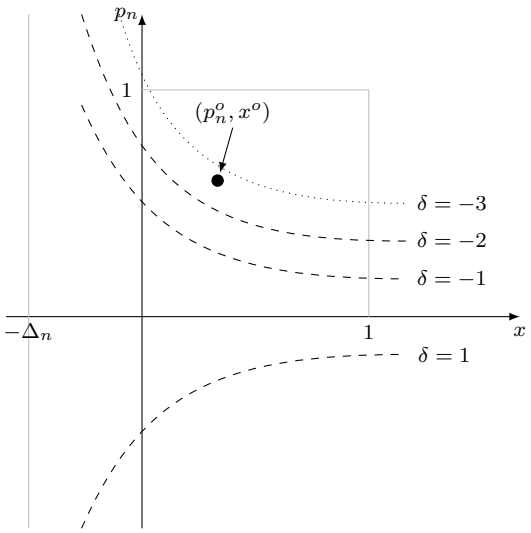

Figure 4.1: The quadratic constraints are partitioned into those below (p_n^o, x^o) that are tight (dashed curves), and those above (p_n^o, x^o) where $p_i = 0$ (dotted curves).

Lemma 5. *Assume $x^o > 0$ and $p_n^o < 1$. Let $p_n^o(x^o + \Delta_n) + \delta = 0$. If for some i, $\delta_{i,n} < \delta$ then $p_i = 0$. If for some i, $\delta_{i,n} > \delta$ then $p_n^o(x^o + \Delta_n) + \delta_{i,n} = p_i(x^o + \Delta_i)$. If for some i, $\delta_{i,n} = \delta$ then $p_i = 0$ and $p_n^o(x^o + \Delta_n) + \delta_{i,n} = p_i(x^o + \Delta_i)$.*

Proof. The quadratic constraint for p_i is $p_n^o(x^o + \Delta_n) + \delta_{i,n} \leq p_i(x^o + \Delta_i)$. By Lemma 4, either $p_i = 0$ or the constraint is tight. If $p_n^o(x^o + \Delta_n) + \delta_{i,n} < 0$, then, since $p_i \geq 0$ and $x^o + \Delta_i \geq 0$, the constraint cannot be tight. Hence, $p_i = 0$. If $p_n^o(x^o + \Delta_n) + \delta_{i,n} > 0$, then, $p_i \neq 0$ or else with $p_i = 0$ the constraint is not satisfied. Hence the constraint is tight. The last case with $p_n^o(x^o + \Delta_n) + \delta_{i,n} = 0$ is trivial. □

From Lemma 5, if p_n^o, x^o lies in the region between the adjacent hyperbolas given by $p_n^o(x^o + \Delta_n) + \delta_{i,n} = 0$ and $p_n^o(x^o + \Delta_n) + \delta_{j,n} = 0$ (and $0 < x^o \leq 1$ and $0 \leq p_n^o < 1$), then $\delta_{i,n} \leq 0$ and $p_i \geq 0$ and for the k^{th} quadratic constraint with $\delta_{k,n} < \delta_{i,n}$, $p_k = 0$ and for the j^{th} quadratic constraint with $\delta_{j,n} > \delta_{i,n}$, $p_j \neq 0$ and the constraint is tight.

These insights lead to Algorithm 3. After handling the case of $x = 0$ and $p_n = 1$ separately, the algorithm sorts the δ's to get $\delta_{(1),n}, \ldots, \delta_{(n-1),n}$ in ascending order. Then, it iterates over the sorted δ's until a non-negative δ is reached, assuming the corresponding p_i's to be zero and the other quadratic constraints to be equalities, and using the subroutine EQ_OPT to solve the induced sub-problem. For ease of

exposition we assume δ's to be distinct, but the extension to repeated δ's is quite natural and does not require any new results. The sub-problem for the i^{th} iteration is given by the problem $Q_{n,i}$:

$$\begin{aligned}
\max_{x,p_{(1)},\ldots,p_{(i)},p_n} \quad & p_n\Delta_{D,n} - ax \ , \\
\text{subject to} \quad & p_n(x+\Delta_n) + \delta_{(i),n} \geq 0 \ , \\
& \text{if } i \geq 2 \text{ then } p_n(x+\Delta_n) + \delta_{(i-1),n} < 0 \ , \\
& \forall j \geq i.\ p_n(x+\Delta_n) + \delta_{(j),n} = p_{(j)}(x+\Delta_j) \ , \\
& \forall j > i.\ 0 < p_{(j)} \leq 1 \ , \\
& 0 \leq p_{(i)} \leq 1 \ , \ \textstyle\sum_{k=i}^{n-1} p_{(k)} = 1 - p_n \ , \ 0 \leq p_n < 1 \ , \\
& 0 < x \leq 1 \ .
\end{aligned}$$

The best (maximum) solution from all the sub-problems (including $x = 0$ and $p_n = 1$) is chosen as the final answer.

Lemma 6. *Assuming* EQ_OPT *produces an ϵ-additive approximate objective value, Algorithm 3 finds an ϵ-additive approximate objective of optimization problem P_n.*

The proof is present in Appendix B.5.

EQ_OPT solves a two-variable problem $R_{n,i}$ instead of $Q_{n,i}$. The problem $R_{n,i}$ is defined as follows:

$$\begin{aligned}
& \max_{x,p_n} \quad p_n\Delta_{D,n} - ax \ , \\
& \text{subject to} \\
& p_n(x+\Delta_n) + \delta_{(i),n} \geq 0 \ , \\
& \text{if } i \geq 2 \text{ then } p_n(x+\Delta_n) + \delta_{(i-1),n} < 0 \ , \\
& p_n\left(1 + \textstyle\sum_{j:i\leq j\leq n-1} \frac{x+\Delta_n}{x+\Delta_{(j)}}\right) = 1 - \sum_{j:i\leq j\leq n-1} \frac{\delta_{(j),n}}{x+\Delta_{(j)}} \ , \\
& 0 \leq p_n < 1 \ , \\
& 0 < x \leq 1 \ .
\end{aligned}$$

The following result justifies solving $R_{n,i}$ instead of $Q_{n,i}$.

Lemma 7. *$Q_{n,i}$ and $R_{n,i}$ are equivalent for all i.*

Proof. Since the objectives of both problems are identical, we prove that the feasible regions for the variables in the objective (p_n, x) are identical. Assume $p_n, x, p_{(i)}, \ldots, p_{(n-1)}$ is feasible in $Q_{n,i}$. The first two constraints are the same in $Q_{n,i}$ and $R_{n,i}$. Divide each equality quadratic constraint corresponding to non-zero $p_{(j)}$ by $x + \Delta_{(j)}$. Add

Algorithm 3: APX_SOLVE(ϵ, P_n)

$l \leftarrow prec(\epsilon, n, K)$, where $prec$ is defined after Lemma 10

Sort δ's in ascending order to get $\delta_{(1),n}, \ldots, \delta_{(n-1),n}$, with corresponding variables $p_{(1)}, \ldots, p_{(n-1)}$ and quadratic constraints $C_{(1)}, \ldots, C_{(n-1)}$

Solve the LP problem for the two cases when $x = 0$ and $p_n = 1$ respectively. Let the solution be $S^0, p^0_{(1)}, \ldots, p^0_{(n-1)}, p^0_n, x^0$ and $S^{-1}, p^{-1}_{(1)}, \ldots, p^{-1}_{(n-1)}, p^{-1}_n, x^{-1}$ respectively.

for $i \leftarrow 1$ **to** $n-1$ **do**

  if $\delta_{(i),n} \leq 0 \vee (\delta_{(i),n} > 0 \wedge \delta_{(i-1),n} < 0)$ **then**

    $p_{(j)} \leftarrow 0$ for $j < i$.

    Set constraints $C_{(i)}, \ldots, C_{(n-1)}$ to be equalities.

    $S^i, p^i_{(1)}, \ldots, p^i_{(n-1)}, p^i_n, x^i \leftarrow$ EQ_OPT(i, l)

  else

    $S^i \leftarrow -\infty$

$f \leftarrow \arg\max_i \{S^{-1}, S^0, S^1, \ldots, S^i, \ldots, S^{n-1}\}$

$p^f_1, \ldots, p^f_{n-1} \leftarrow$ Unsort $p^f_{(1)}, \ldots, p^f_{(n-1)}$

return $p^f_1, \ldots, p^f_n, x^f$

all such constraints to get:

$$-\sum_{j:1\leq j\leq i} p_{(j)} + p_n \left(\sum_{j:1\leq j\leq i} \frac{x+\Delta_n}{x+\Delta_{(j)}} \right) + \sum_{j:1\leq j\leq i} \frac{\delta_{(j),n}}{x+\Delta_{(j)}} = 0$$

Then, since $\sum_{k:1\leq k\leq i} p_{(k)} = 1 - p_n$ we get

$$p_n \left(1 + \sum_{j:i\leq j\leq n-1} \frac{x+\Delta_n}{x+\Delta_{(j)}} \right) = 1 - \sum_{j:i\leq j\leq n-1} \frac{\delta_{(j),n}}{x+\Delta_{(j)}}.$$

The last two constraints are the same in $Q_{n,i}$ and $R_{n,i}$.

Next, assume p_n, x is feasible in $R_{n,i}$. Choose

$$p_{(j)} = p_n \left(\frac{x+\Delta_n}{x+\Delta_{(j)}} \right) + \frac{\delta_{(j),n}}{x+\Delta_{(j)}} \ .$$

Since $p_n(x+\Delta_n)+\delta_{(i),n} \geq 0$, we have $p_{(i)} \geq 0$, and since $p_n(x+\Delta_n)+\delta_{(j),n} > 0$ for $j > i$ (δ's are distinct) we have $p_{(j)} > 0$. Also,

$$\sum_{j=i}^{n-1} p_{(j)} = p_n \left(\sum_{j:i\leq j\leq n-1} \frac{x+\Delta_n}{x+\Delta_{(j)}} \right) + \sum_{j:i\leq j\leq n-1} \frac{\delta_{(j),n}}{x+\Delta_{(j)}},$$

which by the third constraint of $R_{n,i}$ is $1-p_n$. This implies $p_{(j)} \leq 1$. Thus, $p_n, x, p_{(i)}, \ldots, p_{(n-1)}$ is feasible in $Q_{n,i}$.

□

The equality constraint in $R_{n,i}$, which forms a curve K_i, allows substituting p_n with a function $F_i(x)$ of the form $f(x)/g(x)$. Then, the steps in EQ_OPT involve taking the derivative of the objective $f(x)/g(x)$ and finding those roots of the derivative that ensure that x and p_n satisfy all the constraints. The points with zero derivative are however local maxima only. To find the global maxima, other values of x of interest are where the curve K_i intersects the *closed* boundary of the region defined by the constraints. Only the closed boundaries are of interest, as maxima (rather suprema) attained on open boundaries are limit points that are not contained in the constraint region. However, such points are covered in the other optimization problems, as shown below.

The limit point on the open boundary $p_n(x+\Delta_n)+\delta_{(i-1),n} < 0$ is given by the roots of $F_i(x) + \frac{\delta_{(i-1),n}}{x+\Delta_n}$. This point is the same as the point considered on the closed boundary $p_n(x+\Delta_n)+\delta_{(i-1),n} \geq 0$ in problem $R_{n,i-1}$ given by roots of $F_{i-1}(x) + \frac{\delta_{(i-1),n}}{x+\Delta_n}$, since $F_{i-1}(x) = F_i(x)$ when $p_n(x+\Delta_n)+\delta_{(i-1),n} = 0$. Also, the other cases when $x = 0$ and $p_n = 1$ are covered by the LP solved at the beginning of Algorithm 3.

The closed boundary in $R_{n,i}$ are obtained from the constraint $p_n(x+\Delta_n)+\delta_{(i),n} \geq 0$, $0 \leq p_n$ and $x \leq 1$. The value x of the intersection of $p_n(x+\Delta_n)+\delta_{(i),n} = 0$ and K_i is given by the roots of $F_i(x) + \frac{\delta_{(i),n}}{x+\Delta_n} = 0$. The value x of the intersection of $p_n = 0$ and K_i is given by roots of $F_i(x) = 0$. The value x of the intersection of $x = 1$ and K_i is simply $x = 1$. Additionally, as checked in EQ_OPT, all these intersection points must lie with the constraint regions defined in $Q_{n,i}$.

The optimal x is then the value among all the points of interest stated above that yields the maximum value for $\frac{f(x)}{g(x)}$. Algorithm 4 describes EQ_OPT, which employs a root finding subroutine ROOTS. Algorithm 4 also takes care of approximate results

Algorithm 4: EQ_OPT(i, l)

Define $F_i(x) = \frac{1-\sum_{j:1\le j\le i-1} \frac{\delta_{j,n}}{x+\Delta_j}}{1+\sum_{j:1\le j\le i-1} \frac{x+\Delta_n}{x+\Delta_j}}$

Define $feas(x) = \begin{cases} true & (x, F_i(x)) \text{ is feasible for } R_{n,i} \\ false & \text{otherwise} \end{cases}$

Find polynomials f, g such that $\frac{f(x)}{g(x)} = F_i(x)\Delta_{D,n} - ax$

$h(x) \leftarrow g(x)f'(x) - f(x)g'(x)$

$\{r_1, \ldots, r_s\} \leftarrow \mathsf{ROOTS}(h(x), l)$

$\{r_{s+1}, \ldots, r_t\} \leftarrow \mathsf{ROOTS}(F_i(x) + \frac{\delta_{(i),n}}{x+\Delta_n}, l)$

$\{r_{t+1}, \ldots, r_u\} \leftarrow \mathsf{ROOTS}(F_i(x), l)$

$r_{u+1} \leftarrow 1$

for $k \leftarrow 1$ **to** $u+1$ **do**

  if $feas(r_k)$ **then**

    $O_k \leftarrow \frac{f(r_k)}{g(r_k)}$

  else

    if $feas(r_k - 2^{-l})$ **then**

      $O_k \leftarrow \frac{f(r_k-2^{-l})}{g(r_k-2^{-l})}$; $r_k \leftarrow r_k - 2^{-l}$

    else

      if $feas(r_k + 2^{-l})$ **then**

        $O_k \leftarrow \frac{f(r_k+2^{-l})}{g(r_k+2^{-l})}$; $r_k \leftarrow r_k + 2^{-l}$

      else

        $O_k \leftarrow -\infty$

$b \leftarrow \arg\max_k \{O_1, \ldots, O_k, \ldots, O_{u+1}\}$

$p_{(j)} \leftarrow 0$ for $j < i$

$p_{(j)} \leftarrow \frac{p_n(r_b + \Delta_n) + \delta_{(j),n}}{r_b + \Delta_{(j)}}$ for $j \in \{i, \ldots, n-1\}$

return $O_b, p_{(1)}, \ldots, p_{(n-1)}, p_n, r_b$

returned by the ROOTS. As a result of the 2^{-l} approximation in the value of x, the computed x and p_n can lie outside the constraint region when the actual x and p_n are very near the boundary of the region. Thus, we check for containment in the

constraint region for points $x \pm 2^{-l}$ and accept the point if the check passes.

Remark (dummy target): As discussed in Section 4.1, we allow for a dummy target with all costs zero. Let this target be t_0. For n not representing 0, there is an extra quadratic constraint given by $p_0(-x_0 - \Delta_0) + p_n(x + \Delta_n) + \delta_{0,n} \leq 0$, but, as x_0 and Δ_0 are 0 the constraint is just $p_n(x + \Delta_n) + \delta_{0,n} \leq 0$. When n represents 0, then the i^{th} quadratic constraint is $p_i(-x - \Delta_i) + \delta_{i,0} \leq 0$, and the objective is independent of p_n as $\Delta_{D,n} = 0$. We first claim that $p_0 = 0$ at any optimal solution. The proof is provided in Lemma 18 in Appendix. Thus, Lemma 4 and 5 continue to hold for $i = 1$ to $n - 1$ with the additional restriction that $p_n^o(x^o + \Delta_n) + \delta_{0,n} \leq 0$.

Thus, when n does not represent 0, Algorithm 3 runs with the the additional check $\delta_{(i),n} < \delta_{0,n}$ in the if condition inside the loop. Algorithm 4 stays the same, except the additional constraint that $p_0 = 0$. The other lemmas and the final results stay the same. When n represents 0, then x needs to be the smallest possible, and the problem can be solved analytically.

4.2.3 Analysis

Before analyzing the algorithm's approximation guarantee we need a few results that we state below.

Lemma 8. *The maximum bit precision of any coefficient of the polynomials given as input to* ROOTS *is* $2n(K + 1.5) + \log(n)$.

Proof. The maximum bit precision will be obtained in $g(x)f'(x) - f(x)g'(x)$. Consider the worst case when $i = 1$. Then, $f(x)$ is of degree n and $g(x)$ of degree $n - 1$. Therefore, the bit precision of $f(x)$ and $g(x)$ is upper bounded by $nK + \log(\binom{n}{n/2})$, where nK comes from multiplying n K-bit numbers and $\log(\binom{n}{n/2})$ arises from the maximum number of terms summed in forming any coefficient. Thus, using the fact that $\binom{n}{n/2} \leq (2e)^{n/2}$ the upper bound is approximately $n(K + 1.5)$. We conclude that the bit precision of $g(x)f'(x) - f(x)g'(x)$ is upper bounded by $2n(K + 1.5) + \log(n)$. □

We can now use Cauchy's result on bounds on root of polynomials to obtain a lower bound for x. Cauchy's bound states that given a polynomial $a_n x^n + \ldots + a_0$,

any root x satisfies

$$|x| > 1/\left(1 + \max\{|a_n|/|a_0|, \ldots, |a_1|/|a_0|\}\right) .$$

Using Lemma 8 it can be concluded that any root returned by ROOTS satisfies $x > 2^{-4n(K+1.5)-2\log(n)-1}$.

Let $B = 2^{-4n(K+1.5)-2\log(n)-1}$. The following lemma (whose proof is omitted due to lack of space) bounds the additive approximation error.

Lemma 9. *Assume x is known with an additive accuracy of ϵ, and $\epsilon < B/2$. Then the error in the computed $F(x)$ is at most $\epsilon\Psi$, where $\Psi = \frac{Y+\sqrt{Y^2+4X}}{2}$ and*

$$X = \min\Big\{ \sum_{\substack{j:i\le j\le n-1,\\ \delta_{j,n}<0}} \frac{|\delta_{j,n}|}{(B+\Delta_j)^2}, \sum_{\substack{j:i\le j\le n-1,\\ \delta_{j,n}>0}} \frac{2\delta_{j,n}}{(B+\Delta_j)^2} \Big\}$$

$$Y = \min\Big\{ \sum_{\substack{j:i\le j\le n-1,\\ \Delta_n-\Delta_j<0}} \frac{|\Delta_n-\Delta_j|}{(B+\Delta_j)^2}, \sum_{\substack{j:i\le j\le n-1,\\ \Delta_n-\Delta_j>0}} \frac{2(\Delta_n-\Delta_j)}{(B+\Delta_j)^2} \Big\}$$

Moreover, Ψ is of order $O(n2^{(8n(K+1.5)+4\log(n)+K)})$.

We are finally ready to establish the approximation guarantee of our algorithm.

Lemma 10. *Algorithm 3 solves problem P_n with additive approximation term ϵ if*

$$l > \max\{1 + \log(\frac{\Delta_{D,n}\Psi + a}{\epsilon}), 4n(K+1.5) + 2\log(n) + 3\}.$$

Also, as $\log(\frac{\Delta_{D,n}\Psi+a}{\epsilon}) = O(nK + \log(\frac{1}{\epsilon}))$, l is of order $O(nK + \log(\frac{1}{\epsilon}))$.

Proof. The computed value of x can be at most $2\cdot 2^{-l}$ far from the actual value. The additional factor of 2 arises due to the boundary check in EQ_OPT. Then using Lemma 9, the maximum total additive approximation is $2\cdot 2^{-l}\Delta_{D,n}\Psi + 2\cdot 2^{-l}a$. For this to be less than ϵ, $l > 1 + \log(\frac{\Delta_{D,n}\Psi+a}{\epsilon})$. The other term in the max above arises from the condition $\epsilon < B/2$ (this ϵ represents $2\cdot 2^{-l}$) in Lemma 9. □

Observe that the upper bound on ψ is only in terms of n and K. Thus, we can express l as a function of ϵ, n and K—$l = prec(\epsilon, n, K)$.

We still need to analyze the running time of the algorithm. First, we briefly discuss the known algorithms that we use and their corresponding running-time

guarantees. Linear programming can be done in polynomial time using Karmakar's algorithm [65] with a time bound of $O(n^{3.5}L)$, where L is the length of all inputs.

The splitting circle scheme to find roots of a polynomial combines many varied techniques. The core of the algorithm yields linear polynomials $L_i = a_i x + b_i$ (a, b can be complex) such that the norm of the difference of the actual polynomial P and the product $\prod_i L_i$ is less than 2^{-s}, i.e., $|P - \prod_i L_i| < 2^{-s}$. The norm considered is the sum of absolute values of the coefficient. The running time of the algorithm is $O(n^3 \log n + n^2 s)$ in a pointer based Turing machine. By choosing $s = \theta(nl)$ and choosing the real part of those complex roots that have imaginary value less than 2^{-l}, it is possible to obtain approximations to the real roots of the polynomial with l bit precision in time $O(n^3 \log n + n^3 l)$. The above method may yield real values that lie near complex roots. However, such values will be eliminated in taking the maximum of the objective over all real roots, if they do not lie near a real root.

LLL [64] is a method for finding a short basis of a given lattice. This is used to design polynomial time algorithms for factoring polynomials with rational coefficients into irreducible polynomials over rationals. The complexity of this well-known algorithm is $O((n^{12} + n^9(\log |f|)^3)$, when the polynomial is specified as in the field of integers and $|f|$ is the Euclidean norm of coefficients. For rational coefficients specified in k bits, converting to integers yields $\log |f| \simeq \frac{1}{2} \log n + k$. LLL can be used before the splitting circle method to find all rational roots and then the irrational ones can be approximated. With these properties, we can state the following lemma.

Lemma 11. *The running time of Algorithm 3 with input approximation parameter ϵ and inputs of K bit precision is bounded by $O(n^5 K + n^4 \log(\frac{1}{\epsilon}))$. Using LLL yields the running time $O(\max\{n^{13}K^3,\ n^5 K + n^4 \log(\frac{1}{\epsilon})\})$*

Proof. The length of all inputs is $O(nK)$, where K is the bit precision of each constant. The linear programs can be computed in time $O(n^{4.5}K)$. The loop in Algorithm 3 runs less than n times and calls EQ_OPT. In EQ_OPT, the computation happens in calls to ROOTS and evaluation of the polynomial for each root found. ROOTS is called three times with a polynomial of degree less than $2n$ and coefficient bit precision less than $2n(K + 1.5) + \log(n)$. Thus, the total number of roots found is less than $6n$ and the precision of roots is l bits. By Horner's method [66], polynomial evaluation can be done in the following simple manner: given a polyno-

mial $a_n x^n + \ldots + a_0$ to be evaluated at x_0 computing the following values yields the answer as b_0, $b_n = a_n$, $b_{n-1} = a_{n-1} + b_n x_0$, $\ldots$, $b_0 = a_0 + b_1 x_0$. From Lemma 10 we get $l \geq 2n(K + 1.5) + \log(n)$, thus, b_i is approximately $(n + 1 - i)l$ bits, and each computation involves multiplying two numbers with less than $(n+1-i)l$ bits each. We assume a pointer-based machine, thus multiplication is linear in number of bits. Hence the total time required for polynomial evaluation is $O(n^2 l)$. The total time spent in all polynomial evaluation is $O(n^3 l)$. The splitting circle method takes time $O(n^3 \log n + n^3 l)$. Using Lemma 10 we get $O(n^4 K + n^3 \log(\frac{1}{\epsilon}))$ as the running time of EQ_OPT. Thus, the total time is $O(n^5 K + n^4 \log(\frac{1}{\epsilon}))$.

When using LLL, the time in ROOTS in dominated by LLL. The time for LLL is given by $O(n^{12} + n^9(\log n + nK)^3)$, which is $O(n^{12} K^3)$. Thus, the overall the time is bounded by $O(\max\{n^{13} K^3,\ n^4 l)$, which using Lemma 10 is $O(\max\{n^{13} K^3,\ n^5 K + n^4 \log(\frac{1}{\epsilon})\})$. □

4.3 Discussion

We have presented a simple model of audit games, that reveals the hard nature of the technical problem at hand. Modulo the punishment parameter our setting reduces to the simplest model of security games. However, the security game framework is in general much more expressive. The security games model [67] includes a defender that controls multiple security resources, where each resource can be assigned to one of several *schedules*, which are subsets of targets. For example, a security camera pointed in a specific direction monitors all targets in its field of view. As audit games are also applicable in the context of prevention, the notion of schedules is also relevant for audit games. We explore these and other extensions of audit games in the following chapter.

Chapter 5

Audit Game: Multiple Defender Resources

Building on the model in the last chapter, we generalize the model with multiple inspections, and allow for constraints on how the inspections can be used (analogous to schedules in security games [21]). We develop efficient algorithms for such scenarios, and as part of that algorithm also improve upon algorithms for special cases of security games.

Our approach captures a number of practical audit scenarios. Often, even when the total number of targets to be audited is very large, many similar targets can be clubbed together as a type, e,g., celebrity record accesses, finance related issues, etc. The types of targets are typically significantly fewer than the number of targets. Our algorithm utilizes the types to run faster on such instances.

Our model also captures a number of other audit settings: (1) localized auditing, i.e., managers inspect actions of their direct reportees or (2) central auditors that audit targets that the managers are not capable of auditing (all such targets collapse into one node) or (3) a sub-ordinate being managed by a two managers or (4) a constant number of sub-ordinates managed by at-most a constant number of managers or (5) punishment levels chosen specific to targets. These variations capture many realistic scenarios.

We discuss our contributions briefly before describing the model and algorithms in detail.

Our Model

We allow multiple inspections by the auditor, treating each inspection as a resource. This naturally restricts each resource to inspecting one target at most. However, inspection resources may be constrained in the targets that they can potentially inspect; this captures scenarios where only some specialized auditors can investigate certain cases. Examples of specialized auditing include scenarios such as billing related case investigated by financial auditors from billing department, localized auditing structure in which managers audit their direct sub-ordinates and the same localized structure, but, with company wide auditors for cases that some mangers are not capable of auditing. Mathematically, building on the model from the last chapter, we introduce variable p_i^j, the probability of inspection resource j inspecting target i. Some of these variables are forced to be zero, reflecting the specialized auditing scenario. Then, the probability of inspecting a target t_i is $p_i = \sum_j p_i^j$. We also informally present the simple extension to the model with multiple m attacks by the adversary. This extension involves introducing constraints that capture the scenario when the adversary's utility in attacking any of the given m targets is higher than the utility in attacking any of the other $n - m$ targets.

Further, later in the chapter, we extend the model to the scenario where an auditor can choose a different punishment level $(x_1, \ldots, x_n)$ for different targets $t_1, \ldots, t_n$. This extension to the model reflects the common practice in law of proportional punishment [26]. As earlier, our choice of optimal punishment levels is still based on optimizing the utility of the defender, instead of qualitative means employed in law.

Our Results

As in the last chapter, our approach to computing the Stackelberg equilibrium is based on the multiple LPs technique of Conitzer and Sandholm [24]. We demonstrate that the intuitive approach of discretizing the punishment level x and solving the resulting linear programs for each fixed x makes the problem *fixed parameter tractable* (FPT), i.e., the problem can be solved efficiently given the bit precision of the input is considered a constant. The FPT approach can also tackle the problem where the adversary attacks upto m targets, where m is a constant.

Next, we present an approach to reduce the problem size by eliminating many

variables of the optimization problem, namely all the p_i^j's, though at the cost of increasing the number of constraints (in p_i's and x) in the problem. We characterize the conditions under which the number of constraints do not increase exponentially, and hence allow for solving the audit game problem efficiently (FPTAS). These conditions capture many specialized auditing scenarios including all the specialized auditing examples we listed above. We show that it is possible to obtain an FPTAS if the number on constraints are still polynomially many. Further, if the number of such constraint are small they also reduce the running time of the FPT. We demonstrate this improvement in running time experimentally for the FPT approach, achieving improvement of 5x in running time with 200 targets, 10 human auditors with 10 inspections each (100 inspection resources) in a localized auditing structure, where each auditor inspects 20 targets.

Finally, we show that solving the case with multiple punishment levels is also a FPT problem, but not solved in the naive manner by simply discretizing each punishment level. We provide a FPT for this scenario by discretizing only p_i, x_i (for the case when t_i is the target under attack) and reducing the resultant optimization problem to a second order cone program (SOCP), that is known to be solvable efficiently.

5.1 Audit Game Model with Multiple Inspections

The audit game features two players: the defender (D), and the adversary (A). The defender wants to audit n targets $t_1, \ldots, t_n$, but has limited resources which allow for auditing only some of the n targets. The defender employs human auditors, with each human auditor performing a fixed number of inspections. Viewing each inspection of an auditor as a resource, the defender has k inspection resources $\{s_1, \ldots, s_k\}$ at his disposal. An important point to emphasize is that each inspection resource can audit only one target. Also, a human auditor may not have the expertise to audit some of the targets. To capture this idea we define a set of tuples R such that any tuple $(j, i) \in R$ means that inspection resource s_j cannot audit target t_i. Thus, inspections are constrained by the set of targets that they can audit.

A pure action of the defender is to choose the allocation of inspection resources to targets. Such an allocation can be represented by a matrix a where a_i^j is 1 (rows

are resources, columns targets) if resource s_j is allocated to target t_i, otherwise it is 0. A randomized strategy is a matrix of probabilities, with p_i^j the probability of inspection s_j auditing target t_i subject to the following constraints

$$\sum_{j=1}^{k} p_i^j \leq 1, \quad \sum_{i=1}^{n} p_i^j \leq 1 \quad \text{for all } i, j \text{ and}$$
$$p_i^j = 0 \text{ for all } (j,i) \in R \text{ and } \forall (j,i).\ p_i^j \geq 0 \quad .$$

We call these constraints the grid constraints, as they can be represented in a grid as follows (with p_i's the probability of inspecting target t_i):

	t_1	...	...	t_n	
s_1	p_1^1	...	...	p_n^1	$\sum_i p_i^1 \leq 1$
...	...	...	...	...	...
...	...	...	...	...	...
s_k	p_1^k	...	...	p_n^k	$\sum_i p_i^k \leq 1$
	$p_1 = \sum_j p_1^j \leq 1$	...	...	$p_n = \sum_j p_n^j \leq 1$	

This choice of randomized strategy is justified by the fact that such a matrix can be decomposed into pure actions efficiently (see below for details). The adversary attacks one target such that given the defender's strategy the adversary's choice of violation is the best response.

For the sake of completeness, we state the Birkhoff-von Neumann theorem from Korzhyg et al. [31], that is used to decompose the probability matrix into pure actions efficiently.

(Birkhoff-von Neumann [68]). Consider an $m \times n$ matrix M with real numbers $a_{ij} \in [0,1]$, such that for each $1 \leq i \leq m$, $\sum_{j=1}^{n} a_{ij} \leq 1$, and for each $1 \leq j \leq n$, $\sum_{i=1}^{m} a_{ij} \leq 1$. Then, there exist matrices $M_1, M_2, \ldots, M_q$, and weights $w_1, w_2, \ldots, w_q \in (0,1]$, such that:

1. $\sum_{k=1}^{q} w_k = 1$
2. $\sum_{k=1}^{q} w_k M_k = M$
3. *for each $1 \leq k \leq q$, the elements of M_k are $a_{ij}^k \in \{0,1\}$*

4. *for each* $1 \le k \le q$, *we have: for each* $1 \le i \le m$, $\sum_{j=1}^{n} a_{ij}^{k} \le 1$, *and for each* $1 \le j \le n$, $\sum_{i=1}^{m} a_{ij}^{k} \le 1$.

Moreover, q *is* $O((m+n)^2)$, *and the* M_k *and* w_k *can be found in* $O((m+n)^{4.5})$ *time using Dulmage-Halperin algorithm [69].*

Clearly, our variables p_i^j's can be considered as the matrix M in the result above, and hence can be decomposed into pure actions efficiently.

We follow notational convention similar to earlier papers on security games: let the utility of the defender be $U_D^a(t_i)$ when audited target t_i was found to be attacked, and $U_D^u(t_i)$ when unaudited target t_i was found to be attacked. The attacks (violation) on unaudited targets are discovered by an external source. Similarly, define the utility of the attacker as $U_A^a(t_i)$ when the attacked target t_i is audited, and $U_A^u(t_i)$ when attacked target t_i is not audited, excluding any punishment imposed by the defender. Attacks discovered externally are costly for the defender, thus, $U_D^a(t_i) > U_D^u(t_i)$. Similarly, attacks not discovered by internal audits are more beneficial to the attacker, and $U_A^u(t_i) > U_A^a(t_i)$.

Following the naming convention of Korzhyk, et al. [31], we have a game model with heterogeneous resources and singleton schedules, but with a critical difference. The additional component in audit games that makes the problem technically challenging is punishments. The defender chooses a punishment "rate" $x \in [0, 1]$ such that if auditing detects an attack, the attacker is fined an amount x. However, punishment is not free—the defender incurs a cost for punishing, e.g., for creating a fearful environment. For ease of exposition, we model this cost as a linear function ax, where $a > 0$; however, our results directly extend to any cost function polynomial in x. Assuming $x \in [0, 1]$ is also without loss of generality as utilities can be scaled to be comparable to x. We do assume the punishment rate is fixed and deterministic; a natural assumption that corresponds to a consistent policy.

We can now define the full utility functions. The probabilities $p_1, \ldots, p_n$ of each target being audited is given by $p_i = \sum_{j=1}^{k} p_i^j$. Then, the utility of the defender when target t_* is attacked is

$$p_* U_D^a(t_*) + (1 - p_*) U_D^u(t_*) - ax.$$

The defender pays a fixed cost ax regardless of the outcome. In the same scenario,

the utility of the attacker when target t_* is attacked is

$$p_*(U_A^a(t_*) - x) + (1 - p_*)U_A^u(t_*).$$

The attacker suffers the punishment x only when attacking an audited target.

Equilibrium. The Stackelberg equilibrium solution involves a commitment by the defender to a strategy (with a possibly randomized allocation of the resource), followed by the best response of the adversary. The mathematical problem involves solving multiple optimization problems, one each for the case when attacking t_* is in fact the best response of the adversary. Thus, assuming t_* is the best response of the adversary, the $*^{th}$ optimization problem P_* in audit games is

$$\begin{array}{ll}
\max\limits_{p_{ij},x} & p_* U_D^a(t_*) + (1 - p_*)U_D^u(t_*) - ax \ , \\
\text{subject to} & \forall i \neq *.\ p_i(U_A^a(t_i) - x) + (1 - p_i)U_A^u(t_i) \\
& \qquad\qquad \leq p_*(U_A^a(t_*) - x) + (1 - p_*)U_A^u(t_*) \ , \\
& \forall j.\ 0 \leq \sum_{i=1}^{n} p_i^j \leq 1 \ , \\
& \forall i.\ 0 \leq p_i = \sum_{j=1}^{k} p_i^j \leq 1 \ ,\ \forall (j,i).\ p_i^j \geq 0 \ , \\
& \forall (j,i) \in R.\ p_i^j = 0 \ , \quad 0 \leq x \leq 1 \ .
\end{array}$$

The first constraint verifies that attacking t_* is indeed a best response. The auditor then solves the n problems $P_1, \ldots, P_n$ (which correspond to the cases where the best response is $t_1, \ldots, t_n$, respectively), and chooses the best solution among all these solutions to obtain the final strategy to be used for auditing. This is a generalization of the multiple LPs approach of Conitzer and Sandholm [24].

Inputs. The inputs to the above problem are specified in K bit precision. Thus, the total length of all inputs is $O(nK)$.

Multiple Attacks. The adversary attacking multiple targets can be modeled as a simple extension, which we state informally. Suppose the attacker attacks m targets. Then, the objective of the optimization must account for the utility when these m targets are attacked; the objective is then a linear function of corresponding probabilities of attacking the m targets and x. Then, add additional constraints to enforce the condition that the m targets are most attractive to the attacker, i.e., the utility of the attacker in attacking any of the m targets is higher than the utility in attacking any of the other $n - m$ targets. The Stackelberg equilibrium would then

need to solve $\binom{n}{m}$ such optimization problems, which is polynomial in number when m is a constant. Also, the case of at-most m attacks will require solving $\sum_{i=1}^{m}\binom{n}{i}$ optimization problems, which again is polynomially many.

5.2 Algorithms

We first describe an intuitive approach to the solve the problem. This approach proves that the problem is a fixed parameter tractable problem.

5.2.1 Fixed Parameter Tractable Problem.

Our first algorithmic result is as follows:

Lemma 12. *The intuitive approach of discretizing x yields a FPT (fixed parameter tractable) algorithm for problem P_n, when the bit precision is considered a constant and the smallest value of x is restricted to be any constant greater than* 0.

The proof is in Appendix C.8.

The optimization problem for any fixed value of x is a linear programming problem. Obtaining an additive approximation of ϵ requires discretization intervals of size $\theta(\epsilon)$ (See Appendix C.8). Thus, the running time is $O(\mathsf{LP}(n)/\epsilon)$, where $\mathsf{LP}(n)$ is the time to solve a linear program with n variables. The result is in FPT only when the bit precision is fixed because the running time is exponential in the bit precision K that is used to specify the input. We note that a binary search on x does not work as the solution is not single-peaked in x, thus, we cannot obtain a FPTAS using a binary search over x. See Section 5.4.2 for the counterexample.

This FPT approach can also deal with scenario in which attackers attacks multiple targets bounded by a constant. Suppose the attacker attacks m targets. Discretizing x would yield linear programs for each fixed value of x. As m is a constant the size of these linear programs is within a constant factor of the single attack FPT.

In the last chapter, an FPTAS algorithm was proposed to compute equilibrium in audit games for the simple case of one inspection. We extend this approach to solve audit games with multiple inspections and single attack in fully polynomial time, but with certain restrictions. We first transform the optimization problem into another equivalent problem with much fewer variables, but more constraints. We

obtain a FPTAS when the set of new constraints obtained by the transformation is polynomial is size.

We also claim that this transformation reduces the running time (by a constant factor) of the FPT algorithm, as the transformation leads to a huge reduction in the number of variables of the problem. We describe this transformation before discussing the algorithm and conditions under which polynomially many new constraints are produced.

5.2.2 Extracting constraints for p_i's

The transformation eliminates variables p_i^j's and instead extracts inequalities (constraints) for the variables p_i's from the constraints below (referred to as grid constraints)

$$\forall j.\ 0 \leq \sum_{i=1}^{n} p_i^j \leq 1\ , \forall i.\ 0 \leq p_i = \sum_{j=1}^{k} p_i^j \leq 1\ ,$$
$$\forall (j,i).\ p_i^j \geq 0\ , \forall (j,i) \in R.\ p_i^j = 0\ .$$

Consider any subset of inspection resources L with the resources in L indexed by $s_1, \ldots, s_{|L|}$ ($|L| \leq k$). Suppose, the set of targets M can only be audited by L. Let M be indexed by $t_1, \ldots, t_{|M|}$. Then, in case $|L| < |M|$, we obtain a constraint $p_{t_1} + \ldots + p_{t_{|M|}} \leq |L|$. Call such a constraint c_{ML}. Consider the set of all such constraints

$$C = \{c_{ML} \mid L \in 2^S, M \in 2^T \text{ audited by } L \text{ only}, |L| < |M|\}$$

where $S = \{s_1, \ldots, s_k\}$ and $T = \{t_1, \ldots, t_n\}$.

Then, we claim the following

Lemma 13. *The optimization problem P_* is equivalent to the optimization problem obtained by replacing the grid constraints by $C \cup \{0 \leq p_i \leq 1\}$.*

The proof is in Appendix C.7.

Observe that the definition of C is constructive and provides an algorithm to obtain C. However, the algorithm has running time complexity $\Omega(2^k)$, whereas, in some scenarios the size of C may be polynomial in k. We return to this issue later in Section 5.2.4, where we explore alternative ways of extracting the constraints in C. For now, we present the FPTAS assuming that the set C has polynomially many constraints.

5.2.3 Optimization Algorithm

We present a FPTAS under the condition that the set C has polynomially many constraints.

FPTAS. Our algorithm builds on our earlier algorithm from the last chapter for the restricted auditing scenario. We solve the problem after extracting constraints C. Thus, the variables in our problem are just p_i's and x. The final solution involves computing feasible values of p_i^j's (this can be done in polynomial time since it is a linear feasibility problem), followed by the Birkhoff-von Neumann decomposition (see details shown in Section 5.1) to obtain the distribution over pure allocations.

First, we simplify our problem P_* by eliminating variables p_i^j and introducing constraints in C, and (wlog) letting $*$ be n.

$$\begin{array}{ll} \max\limits_{p_i,x} & p_n U_D^a(t_n) + (1-p_n)U_D^u(t_n) - ax \ , \\ \text{subject to} & \forall i \neq n.\ p_i(U_A^a(t_i) - x) + (1-p_i)U_A^u(t_i) \\ & \qquad \leq p_*(U_A^a(t_n) - x) + (1-p_n)U_A^u(t_n) \ , \\ & \forall i.\ 0 \leq p_i \leq 1 \ , \qquad 0 \leq x \leq 1 \ , \\ & c \in C \ . \end{array}$$

We rewrite P_n in a more compact form. Let $\Delta_{D,i} = U_D^a(t_i) - U_D^u(t_i)$, $\Delta_i = U_A^u(t_i) - U_A^a(t_i)$ and $\delta_{i,j} = U_A^u(t_i) - U_A^u(t_j)$. $\Delta_{D,i}$ and Δ_i are always positive, and P_n reduces to:

$$\begin{array}{ll} \max\limits_{p_i,x} & p_n\Delta_{D,n} - ax \ , \\ \text{subject to} & \forall i \neq n.\ p_i(-x - \Delta_i) + p_n(x + \Delta_n) + \delta_{i,n} \leq 0 \\ & c \in C, \quad \forall i.\ 0 \leq p_i \leq 1 \ , \quad 0 \leq x \leq 1 \ . \end{array}$$

Property of an optimal point. First, observe that the quadratic inequality can be rewritten as

$$\frac{p_n(x + \Delta_n) + \delta_{i,n}}{x + \Delta_i} \leq p_i \ .$$

Suppose p_i^*'s and x^* is an optimal point, and suppose for some j we have the strict inequality

$$\frac{p_n^*(x^* + \Delta_n) + \delta_{j,n}}{x^* + \Delta_j} < p_j^* \ .$$

Let $p'_j = \min(0, \frac{p^*_n(x^*+\Delta_n)+\delta_{j,n}}{x^*+\Delta_j})$. Then, we claim that $p^*_1, \ldots, p^*_{j-1}, p'_j, p^*_{j+1}, p^*_n, x^*$ is also an optimal point. This is easy to see since decreasing p_j is not restricted by any inequality other than the quadratic inequality and the objective only depends on p_n. As a result, we can restrict the problem to be equalities for all those quadratic constraints for which $p^*_n(x^* + \Delta_n) + \delta_{j,n} \geq 0$, and restrict $p_j = 0$ in case $p^*_n(x^* + \Delta_n) + \delta_{j,n} < 0$.

Decomposing into sub-problems. Then, similar to the approach in Chapter 4, we sort $\delta_{i,n}$'s to get a sorted array $\delta_{(i),n}$ in ascending order. Then, split the optimization problem P_n into sub-problems $EQ_{(j)}$, where in each problem $EQ_{(j)}$ it is assumed that p_n, x lies between the hyperbolas $p_n(x + \Delta_n) + \delta_{(j),n}$ (open boundary) and $p_n(x + \Delta_n) + \delta_{(j+1),n}$ (closed boundary) in the plane spanned by p_n, x. Thus, in $EQ_{(j)}$, $p_n(x + \Delta_n) + \delta_{(j),n}$ is non-negative for all $(k) > (j)$ and negative otherwise. For the negative case, using the property of the optimal point above, we claim that for all $(k) \leq (j)$ we can set $p_{(k)} = 0$. The optimal value for P_n can be found by solving each $EQ_{(j)}$ and taking the best solution from these sub-problems.

The optimization problem for $EQ_{(j)}$:

$$
\begin{array}{ll}
\max\limits_{p_i, x} & p_n \Delta_{D,n} - ax \ , \\
\text{subject to} & \forall (i) > (j).\ 0 \leq \frac{p_n(x+\Delta_n)+\delta_{(i),n}}{x+\Delta_{(i)}} = p_{(i)} \leq 1 \\
& p_n(x + \Delta_n) + \delta_{(j),n} < 0 \\
& p_n(x + \Delta_n) + \delta_{(j+1),n} \geq 0 \\
& \forall (i) \leq (j).\ p_{(i)} = 0 \ , \qquad 0 \leq x \leq 1 \\
& c \in C \ .
\end{array}
$$

As no p_i (except p_n) shows up in the objective, and due to the equality constraints on particular p_i's, we can replace those p_i's by a function of p_n, x. Other p_i's are zero. Then, denote by $c(p_n, x)$ the inequality obtained after substituting p_i with the function of p_n, x (or zero) for constraints $c \in C$. Thus, we get the following two

variable optimization problem

$$\begin{array}{ll} \max\limits_{p_i,x} & p_n\Delta_{D,n} - ax \ , \\ \text{subject to} & \forall (i) > (j).\ \frac{p_n(x+\Delta_n)+\delta_{(i),n}}{x+\Delta_{(i)}} \leq 1 \\ & p_n(x+\Delta_n) + \delta_{(j),n} < 0 \\ & p_n(x+\Delta_n) + \delta_{(j+1),n} \geq 0 \\ & 0 \leq p_n \leq 1 \ , \quad 0 \leq x \leq 1 \\ & c(p_n, x) \in C \ . \end{array}$$

Observe that we removed the $0 \leq$ condition in the first set of constraints because that is implied by the next two constraints. Next, note that any constraint (indexed by b) with two variables p_n, x can be expressed as $p_n \leq f_b(x)$ (closed boundary) for constraint-specific ratio of polynomials $f_b(x)$ or by $p_n < \frac{-\delta_{(j),n}}{x+\Delta_n}$ (open boundary). However, we close the open boundary, i.e., $p_n \leq \frac{-\delta_{(j),n}}{x+\Delta_n}$, and solve the problem, returning an infeasible solution in case the solution is on the boundary $p_n = \frac{-\delta_{(j),n}}{x+\Delta_n}$. This is justified by the fact that the curve $p_n = \frac{-\delta_{(j),n}}{x+\Delta_n}$ is included in the other optimization problem $EQ_{(j-1)}$, and would be output by that sub-problem if it indeed is the global maximizer. Thus, we represent the optimization problem as

$$\begin{array}{ll} \max\limits_{p_i,x} & p_n\Delta_{D,n} - ax \ , \\ \text{subject to} & \forall b \in \{1, \ldots, B\}.\ p_n \leq f_b(x) \\ & 0 \leq p_n \leq 1 \ , \quad 0 \leq x \leq 1 \ , \end{array}$$

where B is the total number of constraints described as above.

Solving sub-problem. We first claim that it is easy to solve the problem with $p_n = 1$ or $p_n = 0$ and $x = 0$ or $x = 1$. The fixed x case is obvious, since the problem is a linear program for fixed x. The fixed p_n case is equivalent to minimizing x with constraints that are of the form $f(x) \leq 0$, where f is a polynomial in x. For any polynomial constraint $f(x) \leq 0$, it is possible to approximate the roots of the polynomial with an additive approximation factor of 2^{-l} and hence find the intervals of x that satisfies the constraint $f(x) \leq 0$ within additive approximation factor of 2^{-l}. Doing so for the polynomially many constraints and finding the intersection of intervals yields the minimum value of x with additive approximation factor of 2^{-l} that satisfies all constraints.

Algorithm 5: APX_SOLVE$(l, EQ_{(j)})$

Solve the problem for $p_n = 0, 1$ and $x = 0, 1$
Collect solutions (p_n, x) from the above in M
for $b \leftarrow 1$ **to** B **do**
  Replace $p_n = f_b(x)$ in the objective to get $F(x) = f_b(x)\Delta_{D,n} - ax$
  Take the derivative to get $F'(x) = \frac{\partial F(x)}{\partial x}$
  $R \leftarrow \mathsf{ROOTS}(F'(x), 2^{-l}, (0, 1))$
  $R' \leftarrow \mathsf{MAKEFEASIBLE}(R)$
  From R' obtain set M' of potential solutions (p_n, x)
  $M \leftarrow M \cup M'$
for $(b, b') \in \{(b, b') \mid b \in B, b \in B, b' > b\}$ **do**
  Equate $f_b(x) = f_{b'}(x)$ to get $F(x) = 0$
  $R \leftarrow \mathsf{ROOTS}(F(x), 2^{-l}, (0, 1))$
  $R' \leftarrow \mathsf{MAKEFEASIBLE}(R)$
  From R obtain set M' of potential solutions (p_n, x)
  $M \leftarrow M \cup M'$
$(p_n^*, x^*) \leftarrow \arg\max_M \{p_n \Delta_{D,n} - ax\}$
return (p_n^*, x^*)

Next, we claim that the optimal point cannot be an interior point of the region defined by the constraints, i.e., it cannot be the case that all the inequalities are strict for the optimal point. We can derive an easy contradiction for this scenario. Assume p_n^*, x is optimal. As f_b is continuous in x, if all inequalities are strict, then it is possible to increase the value of optimal p_n^* by a small amount, without violating the constraints. Clearly, this increased value of p_n results in a higher objective value, contradicting the assumption that p_n^*, x^* is optimal.

Hence, given the above claims, the procedure in Algorithm 5 finds the maximum of the optimization under consideration. Algorithm 5 first searches the maximizer on the boundaries (in the first loop) and then searches the maximizer at the intersection of two boundaries (second loop). The roots are found to an additive approximation factor of 2^{-l} in time polynomial in the size of the polynomial and l and similar to our earlier approach [23] in the last chapter, the case of roots lying outside the feasible region (due to approximation) is taken care of by the function MAKEFEASIBLE. The first loop iterates a maximum of n times, and the second loop iterates a maximum of $\binom{n}{2}$ times. Thus, the above claims prove the following result

Theorem 3. *The optimization problem P_n can be solved with an additive approxi-*

mation factor of 2^{-l} *in time polynomial in the input size and* l*, i.e., our algorithm to solve* P_n *is a FPTAS.*

5.2.4 Conditions for Polynomial Number of Constraints

The definition of the constraints set C provides an algorithm to construct the set. However, as discussed earlier, that algorithm is always exponential. We provide an alternate algorithm to find constraints in C and sufficient conditions under which the number of constraints is polynomial in number. First, the following counterexample reveals that in the worst case the number of constraints in C will be super-polynomial. Consider k resources $s_1, \ldots, s_k$ and $2k$ targets $t_1, \ldots, t_{2k}$. Each resource s_i can inspect targets $t_1, \ldots, t_{2i}$. Consider any set of k targets formed by picking $k/2$ target pairs among t_{2i-1}, t_{2i} for any i, except $i = 1$. The number of such sets are $\binom{k-1}{k/2}$. Each such set of targets results in a valid constraint, since the number of resources that can audit these targets in less than k. This is because the first resource only inspects t_1, t_2, and that pair is never chosen. Thus, we get $\binom{k-1}{k/2}$ constraints, which is clearly not polynomial.

The intuition behind the alternate algorithm is that instead of iterating over sets of inspection resources, we could iterate over sets of targets. As a first step, we identify similar targets and merge them. More precisely, any targets that can be audited by the exact same set of inspections are equivalent. The algorithm to determine constraints is described in Algorithm 6.

The algorithm builds an intersection graph from the merged target, i.e., treat every merged target as a node and two nodes are linked if the two sets of inspection resources corresponding to the nodes intersect. The algorithm iterates through every connected induced sub-graph and builds constraints from the targets associated with the nodes in the sub-graphs and the set of inspection resources associated with them.

Next, we prove the correctness of the Algorithm 6. After that, we investigate the conditions under which the Algorithm 6 runs efficiently, and yields a polynomial sized C.

Lemma 14. *Algorithm 6 outputs constraints that defines the same convex polytope in* $p_1, \ldots, p_n$ *as the constraints output by the naive algorithm (iterating over all subsets of resources).*

Proof. Assume constraint c is in the output of the naive algorithm. Restrict the

Algorithm 6: CONSTRAINT_FIND(T, K, R)

Compute F, the map from T to $2^{\{s_1,\ldots,s_k\}}$ using R.
Let $PV(t)$ be the probability variable p_t for target t
Merge targets with same $F(T)$ to get set T' and map W, with $W(t')$ the no. of merged targets that yielded t'
Let $PV(t')$ be the set of probability variables associated with t'
Form an intersection graph G with nodes $t' \in T'$ and each node denoting set $F(T')$
$L \leftarrow$ CONNECTEDSUBGRAPH(G)
$C \leftarrow \phi$
for $l \in L$ **do**
    Let V be all the vertices on l
    $P \leftarrow \cup_{v \in V} PV(v)$
    $k \leftarrow \sum_{v \in W} W(t')$
    if $|P| > k$ **then**
        $C \leftarrow C \cup \{\sum_{p \in P} p \leq k\}$
return C

constraint c to be a non-implied constraint, i.e., c cannot be written as the sum of two or more other constraints in C. Note that such a restriction does not change the region defined by the set C. By the description of the naive algorithm, this constraint must correspond to a set of inspection resources, say S, and the set of targets T_S inspected only by inspection resources in S, and there exists no subset of T_S and S such that the subset of targets is inspected only by the subset of inspection resources. The constraint c is of the form $P(T_S) \leq |S|$, where $P(T_S)$ is the sum of the probability variables for targets in T_S. Now, for the intersection graph representation, every node x represents targets that are inspected by a given subset x_s of inspection resources. For our case, we claim that every $t \in T_S$ is either equivalent to or linked to another target in T_S, otherwise we have the subset $\{t\} \subset S$ inspected only by $t_s \subset T_S$. Thus, the targets in T_S form a connected induced sub-graph. Thus, we conclude that the CONSTRAINT_FIND algorithm will consider this set of targets and find the non-implied constraint c.

Next, assume that CONSTRAINT_FIND finds a constraint c. As this corresponds to a connected induced sub-graph (with targets as nodes), we obtain a set of targets T and a set of inspection resources $\cup_{t \in T} F(t)$ such that targets in T are audited

by $\cup_{t \in T} F(t)$ only. Thus, by definition of the construction of c, this constraint is same as the constraint c' obtained by the naive algorithm when it considers the set $\cup_{t \in T} F(t)$. That is, c will also be found by the naive algorithm. □

The algorithm CONSTRAINT_FIND is clearly not polynomial time in the worst case because of the iteration over all connected sub-graphs. We state here sufficient conditions for CONSTRAINT_FIND to be polynomial. These conditions also correspond to natural auditing scenarios; we discuss this later in Section 5.4.

Lemma 15. CONSTRAINT_FIND *runs in polynomial time if*

- *The intersection graph has* $O(\log n)$ *nodes.*
- *The intersection graph has constant max degree and constant number of nodes with degree* ≥ 3.

Proof.

- Graphs with $O(\log n)$ nodes. The maximum number of connected induced subgraphs in a graph with t nodes is 2^t (take any subset of vertexes). Thus, clearly a graph with $O(\log n)$ nodes will have polynomially many connected sub-graphs.
- Graphs with constant max degree and constant number of nodes with degree ≥ 3. The number of connected induced subgraphs in a tree with max degree d and t vertexes with degree ≥ 3 is bounded from above by $2^{(2(d+1))^{t+1}} n^{(d+1)^{t+1}}$.
  To prove this result, denote by $T(n, d, t)$ the worst case number of connected induced sub-graphs in a graph with n vertices, and max degree d and t vertices with degree ≥ 3.
  Remove a vertex X with degree ≥ 3 to get $k \leq d$ disconnected components. Each connected sub-graph in any component that was linked to X could be combined with any connected sub-graph of any other component linked to X, yielding a new connected sub-graph. Thus, considering every subset of the k components, we get

$$T(n, d, t) \leq 2^k \left(T(n-1, d, t-1)\right)^k$$

  Observing that $k < d+1$, and $T(n-1, d, t-1) \leq T(n, d, t-1)$ we get

$$T(n, d, t) \leq \left(2T(n, d, t-1)\right)^{d+1}$$

Thus, we see that $T(n,d,t) = 2^{(2(d+1))^{t+1}}(n)^{(d+1)^{t+1}}$ satisfies the above equation. As part of the induction, the base case requires reasoning about a graph with max degree either 1 or 2 ($t = 0, d = 1$ or 2). Thus, we need to show that the connected sub-graphs is less than n^{d+1}, which is n^2 for max-degree 1 and n^3 for max-degree 2. The max-degree 1 case is trivial. For the max-degree 2 case, such graphs can be decomposed into paths and cycles, and the number of connected sub-graphs on cycles and paths is less that n^2.

□

Useful in practice. The condition for polynomially many constraints capture a number of practical audit scenarios. Often many similar targets can be clubbed together as a type, e,g., celebrity record accesses, finance related issues, etc. The types of targets are typically significantly fewer than the number of targets. We capture such scenarios in the case when the number of equivalence classes of targets (nodes in the intersection graph) is small.

Another scenario commonly practiced is localized auditing, i.e., managers inspect actions of their direct reportee. Localized auditing does not allow us to collapse targets inspected by different managers into the same equivalent class, because of the different set of inspection resources that each target can be inspected by. As a result, the number of equivalence classes of targets can be still high. However, the inspection resources auditing an equivalence class is disjoint from the inspection resources auditing any other equivalence class. Thus, our representation of the equivalence classes as an intersection graph has no edges. i.e., max degree is zero.

Our algorithm can also handle sever other audit settings: (1) central auditors that audit targets that the managers are not capable of auditing (all such targets collapse into one node) or (2) a sub-ordinate being managed by a two managers (nodes with degree two) or (3) a constant number of sub-ordinates managed by at-most a constant number of managers (constant max degree and constant number of nodes with degree three or more). These variations capture many realistic scenarios.

5.3 Target-specific Punishment

It is natural in many scenarios to consider different punishments for different targets. While this can be modeled as a simple extension to the optimization problem P_n,

solving the optimization problem becomes more challenging. The change to P_n just involves using punishment levels $x_1, \ldots, x_n$ for each target instead of the same punishment x for each target. Then, the new optimization problem PX_n is

$$\begin{array}{ll} \max\limits_{p_i,x} & p_n\Delta_{D,n} - \sum_{j\in\{1,\ldots,n\}} a_j x_j \ , \\ \text{subject to} & \forall i \neq n.\ p_i(-x_i - \Delta_i) + p_n(x_n + \Delta_n) + \delta_{i,n} \leq 0 \\ & \forall j.\ 0 \leq \sum_{i=1}^{n} p_i^j \leq 1 \ , \forall i.\ 0 \leq p_i = \sum_{j=1}^{k} p_i^j \leq 1 \ , \\ & \forall (j,i).\ p_i^j \geq 0 \ , \forall (j,i) \in R.\ p_i^j = 0 \ , \\ & 0 \leq x \leq 1 \ . \end{array}$$

Note that the penalty term for punishment level is now a linear combination of each punishment level.

The naive way to approach the above problem is to discretize each of $x_1, \ldots, x_n$ and solve the resulting LP sub-problems. However, such a discretization of size ϵ, even if yielding a $\theta(\epsilon)$ additive approximation, will run in time $O((1/\epsilon)^n)$, which is not polynomial for constant ϵ. However, we show that it is possible to design an algorithm that runs in time polynomial in ϵ. We do so by discretizing only p_n and x_n, and casting the resulting sub-problem as a second-order cone program (SOCP). SOCP can be solved in polynomial time. Let $S(n)$ denote the running time of the SOCP [32]. Then, we obtain a running time of $O(S(n)(1/\epsilon)^2)$, by showing that such a discretization produces an additive approximation of $\theta(\epsilon)$, given fixed bit precision.

Assume p_i^o's and x_i^o's are an optimal point. Reducing the value of p_n by ϵ_p ($\leq \epsilon$) and x_n by ϵ_n ($\leq \epsilon$) always yields a feasible point, as the quadratic inequality is still satisfied and so are the linear inequalities. The value of ϵ_p and ϵ_n are chosen so that the new values of p_n and x_n lie on the discrete grid for p_n, x_n. Thus, the new feasible point F is p_i^o's for $i = 1$ to $n-1$, x_i^o' for $i = 1$ to $n-1$ and $p_n = p_n^o - \epsilon_p$, $x_n = x_n^o - \epsilon_n$. The objective at this feasible point F is off from the optimal value by a linear combination of ϵ_p, ϵ_n with constant coefficients, which is less than a constant times ϵ. Then, the SOCP with the new values of p_n and x_n yields an objective value at least as high as the feasible point F on the grid. Thus, using our approach, we obtain a maximizer that differs from the optimal only by $\theta(\epsilon)$.

Next, we describe a general SOCP problem.

$$\begin{aligned} \max_{y} \quad & f^T y \ , \\ \text{subject to} \quad & \forall i \in \{1, \ldots, m\}. \ ||A_i y + b_i||_2 \leq c_i^T y + d_i \\ & Fy = g \ . \end{aligned}$$

where the optimization variable is $y \in \mathbb{R}^n$, and all constants are of appropriate dimensions. A special case of the above general problem, which we use, is when A, b is chosen such that $Ay + b = [k \ \ (y_i - y_j - k')]^T$ for some constants k, k', and c, d chosen such that $c^y + d = y_i + y_j + k$. Note that it is always possible to choose such A, b, c, d. Then, the second-order inequality

$$||Ay + b||_2 \leq c^T y + d$$

is same as

$$k^2/4 \leq y_i(y_j + k')$$

Our problem can be cast as a SOCP as follows: we can rewrite the quadratic constraints as follows

$$p_n(x_n + \Delta_n) + \delta_{i,n} \leq p_i(x_i + \Delta_i)$$

Using our discretization approach (discretizing p_n, x only) the LHS of the above inequality is a constant. If the constant is negative we throw out the constraint, since the RHS is always positive. If the constant is positive, we rewrite the constraint as a second-order constraint as described above. The rest of the constraints are linear, and can be rewritten in the equality form by introducing slack variables. Thus, the problem for each fixed value of p_n, x_n is a SOCP, and can be solved efficiently. The claims above prove the following result

Lemma 16. *The method described above is an FPT algorithm for solving PX_n.*

The extension to multiple attacks is immediate; the only change required would be to discrete p_j, x_j for j belonging to the index set of the m targets under consideration. Then, the resultant problem for each fixed values of p_j, x_j for j belonging to the index set of the m targets can be cast as a SOCP similar to the single attack case.

Allowing different punishments for different targets reflects the common law

principle of punishment proportional to the crime [26]. In our case, we allow the defender to choose the best cost optimal punishment levels that deter the adversary.

5.4 Experimental Results

We show the experimental results that demonstrate the usefulness of reducing the problem size by getting rid of the grid constraints. After that, we show the counterexample that demonstrates that we cannot obtain a FPTAS just by a binary search on the punishment parameter x.

5.4.1 Performance Improvement with Reduced Constraint Set

We show our experimental results that provide evidence of the reduction in running time as a result of reducing the problem size by eliminating the grid variables. Our experiments were run on a desktop with quad core 3.2 GHz processor and 8GB RAM. We used Matlab as our programming language, using the inbuilt large scale interior point method implementation of linear programming. Our implementations are not optimized for speed, but only serve to compare the naive approach with our reduced problem size approach. We implemented two FPT algorithm, one with reduced problem size and other with the grid variables and constraints intact. For both the algorithms, we used the exact same problem inputs, utilities were generated randomly from the range $[0, 1]$, a was fixed to 0.01, x was discretized with interval size of 0.05 and it was allowed to vary from 0 to 10 (the upper bound of 10, instead of 1, is not important as utilities can be scaled accordingly).

Our first experiment used 100 targets and 10 inspection resources. The 10 resources were divided into groups of two resources each, with each group of resource capable of inspecting 20 of the targets. The targets that any group of resources were capable of inspecting was disjoint from the target set for any other group. We sampled 10 random instances of the problem. The results are shown in Figure 5.1, with green marks for the reduced problem scenario. Clearly, the reduction in problem size leads to better time performance.

Our second experiment used 200 targets and 100 inspection resources. The 100 resources were divided into groups of 10 resources each, with each group of resource capable of inspecting 20 of the targets. The targets that any group of resources

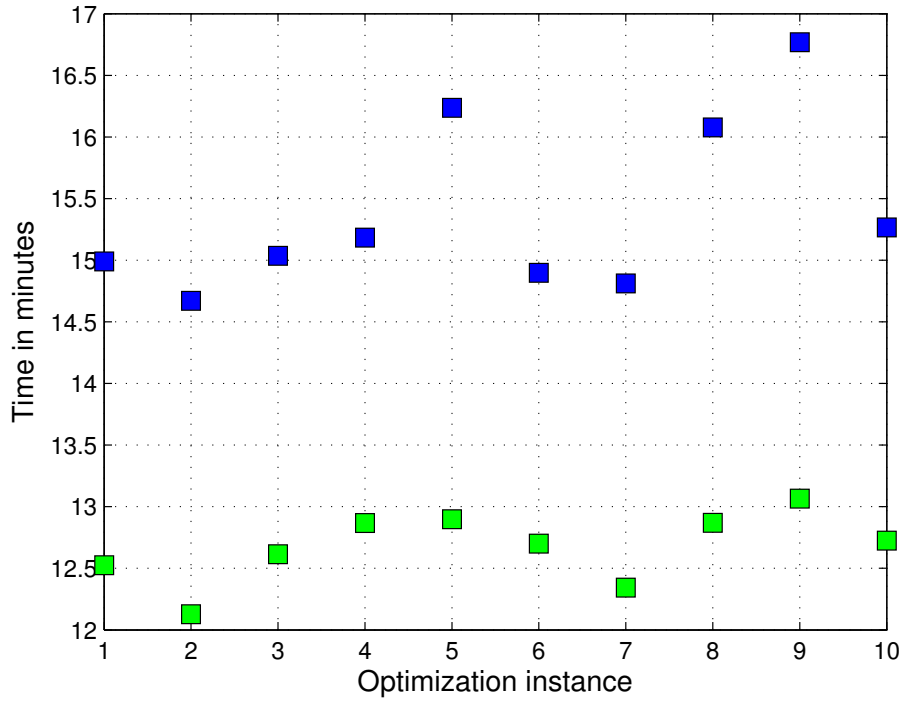

Figure 5.1: FPT algorithm running time with 100 target, 10 inspection resources. Green marks for reduced problem case, blue with the grid constraints

were capable of inspecting was disjoint from the target set for any other group. We sampled 5 random instances of the problem. The results are shown in Figure 5.2, with green marks for the reduced problem scenario. In addition to the reduction in problem size leading to better time performance, we observe that the improvement is much greater for this bigger problem than in the case with 100 targets and 10 resources.

These results suggest that our approach can be used to significantly speed up computing strategies in related security games. As our FPT approach is equivalent to solving the corresponding security game optimization for each discrete value of x, the running time improvements for the FPT algorithm experiments carry over to security games also.

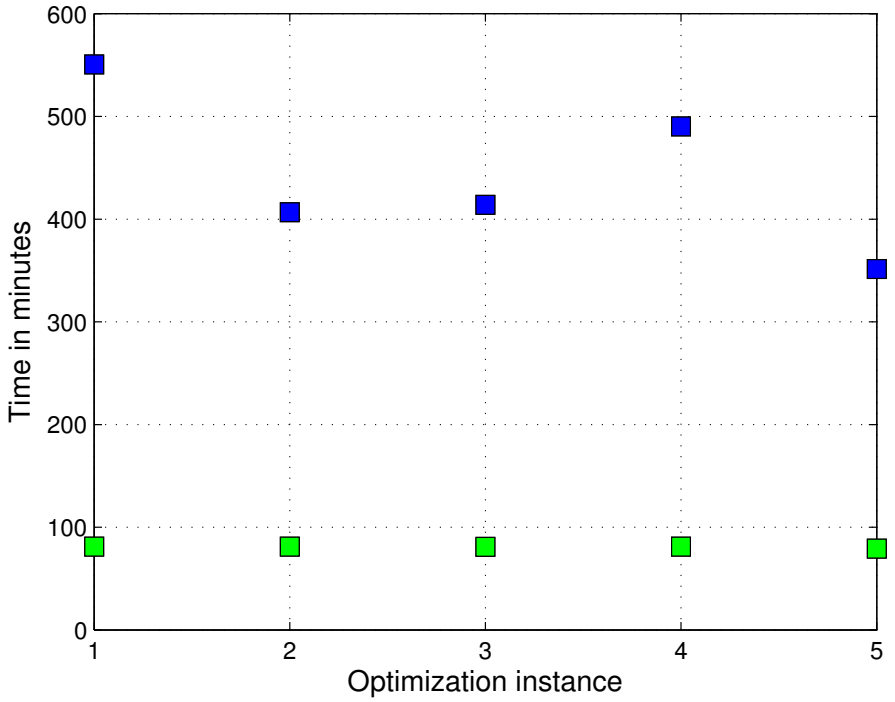

Figure 5.2: FPT algorithm running time with 200 target, 100 inspection resources. Green marks for reduced problem case, blue with the grid constraints.

5.4.2 Maximum Value of Objective is Not Single-peaked

As stated in Section 5.2.1, the solution of the optimization problem is not single peaked in punishment x. Here we show the counterexample that proves this fact. We choose a problem instance with just one defender resource and 7 targets, and we consider only the case when the seventh target is the target under attack. The value of a was chosen to be 0.01, and x was discretized with interval size of 0.005. The various values of utilities are shown in Table 5.1. The variation of maximum utility with x is shown in Figure 5.3. It can be seen that the maximum value is not single peaked in x.

$\mathbf{U_D^a}$	$\mathbf{U_D^u}$	$\mathbf{U_A^a}$	$\mathbf{U_A^u}$
0.614	0.598	0.202	0.287
0.719	0.036	0.869	0.999
0.664	0.063	0.597	0.946
0.440	0.322	0.023	0.624
0.154	0.098	0.899	0.902
0.507	0.170	0.452	0.629
0.662	0.371	1.000	0.999

Table 5.1: Utility values for the counterexample

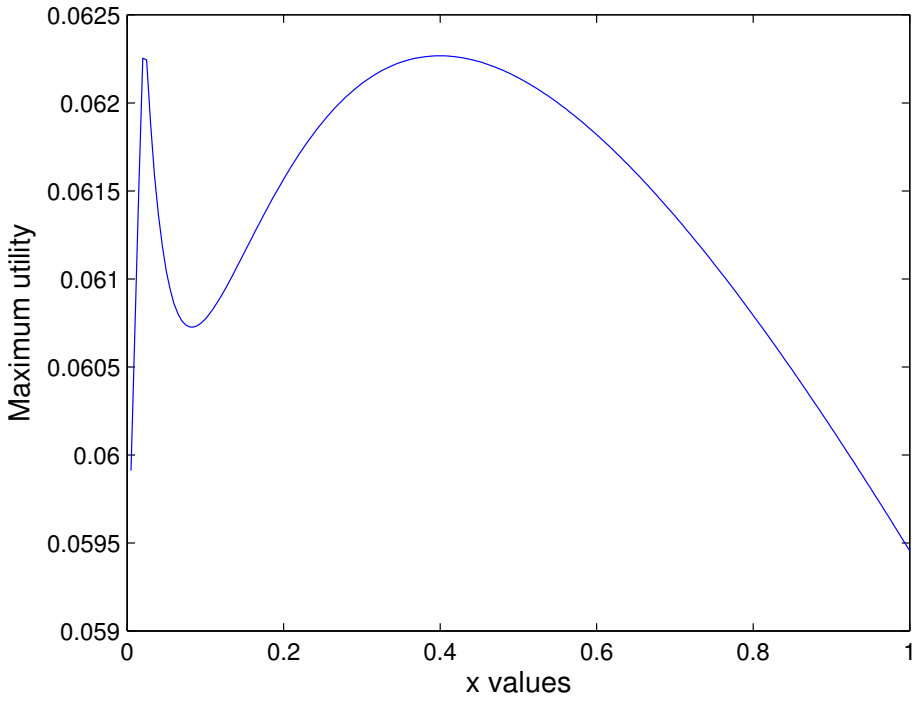

Figure 5.3: Variation of maximum utility with x showing multiple peaks.

Chapter 6

Towards Repeated Audit Games

In the last two chapters on rational adversaries, we have considered one time interaction only, whereas, auditing happens repeatedly. There are multiple ways to approach the repeated setting. Clearly, with myopic adversaries, i.e., adversaries are only concerned about current rewards, repeating the single round interaction over and over is the optimal defender strategy. We demonstrate that such a strategy leads to a *sub-game perfect* equilibrium of the repeated game with any adversary. In addition, we propose a near-rational model of the adversary which models the adversary that act rationally with high probability and in a byzantine manner otherwise (similar to a *trembling hand* assumption [27]). The Stackelberg equilibrium repeated in every round, with the near-rational adversary model, provably yields an *asymmetric approximate subgame perfect equilibrium* (Theorem 4). This equilibrium concept implies that the adversary does not gain at all from deviating from her best response strategy (see Section 6.1). We define this equilibrium concept by adapting the standard notion of approximate sub-game perfect equilibrium, which has a symmetric flavor and permits both players to obtain small gains by unilaterally deviating from their equilibrium strategy. The symmetric equilibrium concept is unsuitable for our security application, where an adversary who deviates motivated by a small gain could cause a big loss for the auditor. Finally, we demonstrate that predictions from our model match real world occurrences, strengthening our claim of practical applicability of our model.

This chapter presents results from our paper [70], skipping lot of details and notation, in order to bring forth the main points from that paper. That model uses

explicit utilities as shown in Chapter 3 in the context of a rational commitment set-up similar to Chapter 4 and 5. The model is slightly more general than the rational setting, where the resources are not fixed, but, can be increased till the limitation set by budget. We present the main results from that paper.

We start by introducing our new equilibrium concepts.

6.1 Equilibrium Concepts

We begin by introducing standard terminology from game theory. We use the following notations in this paper:

- Vectors are represented with an arrow on top, e.g., $\vec{v}$ is a vector. The i^{th} component of a vector is given by $\vec{v}\,(i)$. $\vec{v} \leq \vec{a}$ means that both vectors have the same number of components and for any component i, $\vec{v}\,(i) \leq \vec{a}\,(i)$.
- Random variables are represented in boldface, e.g., $\mathbf{x}$ and $\mathbf{X}$ are random variables.
- $E(\mathbf{X})[q, r]$ denotes the expected value of random variable X, when particular parameters of the probability mass function of $\mathbf{X}$ are set to q and r.

In a one-shot extensive form game (Stackelberg game for two players) players move in order. We assume player 1 moves first followed by player 2. An extensive form repeated game is one in which the round game is a one-shot extensive game. The history is a sequence of actions. Let H be the set of all possible histories. Let S_i be the action space of player i. A strategy of player i is a function $\sigma_i : H_i \to S_i$, where $H_i \subset H$ are the histories in which player i moves. The utility in each round is given by $r_i : S_1 \times S_2 \to \mathbb{R}$. The total utility is a δ_i-discounted sum of utilities of each round, normalized by $1 - \delta_i$.

The definition of strategies extends to extensive form repeated games with public signals. We consider a special case here that resembles our audit game. Player 1 moves first and the action is observed by player 2, then player 2 moves, but, that action may not be perfectly observed, instead resulting in a public signal. Let the space of public signals be Y. In any round, the observed public signal is distributed according to the distribution $\Delta Y(.|s)$, i.e., $\Delta Y(y|s)$ is the probability of seeing signal y when the action profile s is played. In these games, a history is defined as an alternating sequence of player 1'a action and public signals, ending in a public

signal for histories in which player 1 has to move and ending in player 1's move for histories in which player 2 has to move. The actual utility in each round is given by the function $r_i : S_i \times Y \to \mathbb{R}$. The total expected utility g_i is the expected normalized δ_i-discounted sum of utilities of each round, where the expectation is taken over the distribution over public signals and histories. For any history h, the game to be played in the future after h is called the *continuation game* of h with total utility given by $g_i(\sigma, h)$.

A strategy profile (σ_1, σ_2) is a *subgame perfect equilibrium* (SPE) of a repeated game if it is a Nash equilibrium for all continuation games given by any history h [27]. One way of determining if a strategy is a SPE is to determine whether the strategy satisfies the *single stage deviation* property, that is, any *unilateral deviation* by any player *in any single round* is not profitable. We define a natural extension of SPE, which we call *asymmetric subgame perfect equilibrium* (or (ϵ_1, ϵ_2)-SPE), which encompasses SPE as a special case when $\epsilon_1 = \epsilon_2 = 0$.

Definition 1. *((ϵ_1, ϵ_2)-SPE) Denote concatenation operator for histories as ;. Strategy profile σ is a (ϵ_1, ϵ_2)-SPE if for history h in which player 1 has to play, given $h' = h; \sigma_1(h)$ and $h'' = h; s_1$,*

$$\begin{aligned} &E(r_1(\sigma_1(h), \mathbf{y}))[\sigma_1(h), \sigma_2(h')] + \delta_1 E(g_1(\sigma, h'; \mathbf{y}))[\sigma_1(h), \sigma_2(h')] \\ &\geq E(r_1(s_1, \mathbf{y}))[s_1, \sigma_2(h'')] + \delta_1 E(g_1(\sigma, h''; \mathbf{y}))[s_1, \sigma_2(h'')] - \epsilon_1 \end{aligned}$$

for all s_1. For history h in which player 2 has to play, given $a(h)$ is the last action by player 1 in h, for all s_2

$$\begin{aligned} &E(r_2(\sigma_2(h), \mathbf{y}))[a(h), \sigma_2(h)] + \delta_2 E(g_2(\sigma, h; \mathbf{y}))[a(h), \sigma_2(h)] \\ &\geq E(r_2(s_2, \mathbf{y}))[a(h), s_2] + \delta_2 E(g_2(\sigma, h; \mathbf{y}))[a(h), s_2] - \epsilon_2 \end{aligned}$$

We are particularly interested in $(\epsilon_1, 0)$-SPE, where player 1 is the defender and player 2 is the adversary. By setting $\epsilon_2 = 0$, we ensure that a rational adversary will never deviate from the expected equilibrium behavior. Such equilibria are particularly important in security settings, since $\epsilon_2 > 0$ could incentivize the adversary to deviate from her strategy, possibly resulting in significant loss to the defender.

6.2 Game Model

We presented a simplified version of the model in our paper [70]. This rational model uses notation from the Regret Minimizing Audit chapter, and differs from our earlier rational model as the resource cost is also considered in the defender's utility. Also, the utilities are stated explicitly in the manner of Regret Minimizing Audit chapter. However, distinct from RMA, the total utility is a δ-discounted sum of utility in each round, rather than the average utility. This follows the typical convention in rational repeated game theory.

We use the notation and model from Regret Minimizing Audit chapter (Chapter 3). The action space of the defender and adversary is the same, except, an additional action of punishment level $\vec{P}^t$ (each component is the punishment level for each type of violation) for the defender. Then, the defender's utility for round t is

$$\mathbf{L}^t(\langle \vec{a}^t, \vec{v}^t \rangle, \vec{s}^t) = R - R_{int} \cdot \mathbf{O}^t_{int} - R_{ext} \cdot \mathbf{O}^t_{ext} - \vec{C} \cdot \vec{s}^t - \vec{a} \cdot \vec{P}^t \ ,$$

where $\vec{a}$ is a constant (one for each type of violation).

The adversary's utility is given by

$$\mathbf{R}^t(\langle \vec{a}^t, \vec{v}^t \rangle, \langle \vec{s}^t, \vec{P}^t \rangle, \mathbf{O}^t) = \vec{I} \cdot \vec{v}^t - \vec{P}^t \cdot \left(\mathbf{O}^t_{int} + \mathbf{O}^t_{ext} \right) \ .$$

where $\vec{I}$ is the private benefit of the adversary for each type of violation.

We simplify the notation by writing the utilities for each type (removing vector notation, considering R_{int} and R_{ext} as constants), considering only a particular round (dropping superscript t) and also considering expected utility, where $E(\mathbf{O}^t_{int}) = v\nu(\alpha)$ and $E(\mathbf{O}^t_{ext}) = pv(1 - \nu(\alpha))$. Recall that p is the probability of external detection, α stands for s/a and $\nu(\alpha)$ stands for actual efficacy of α fraction inspection. We assume that $\nu(\alpha)$ is $\mu(\alpha)\alpha$, where $\mu(\alpha)$ is a fixed function greater than 1, and $\mu(\alpha)\alpha \leq 1$. Observe that the worst case efficacy is achieved in the case of random sampling of accesses to inspect, for which $\mu = 1$.

The expected utility in each round for the defender is

$$R - R_{int} v(1 - \nu(\alpha)) - R_{ext} v\nu(\alpha) - Cs - aP$$

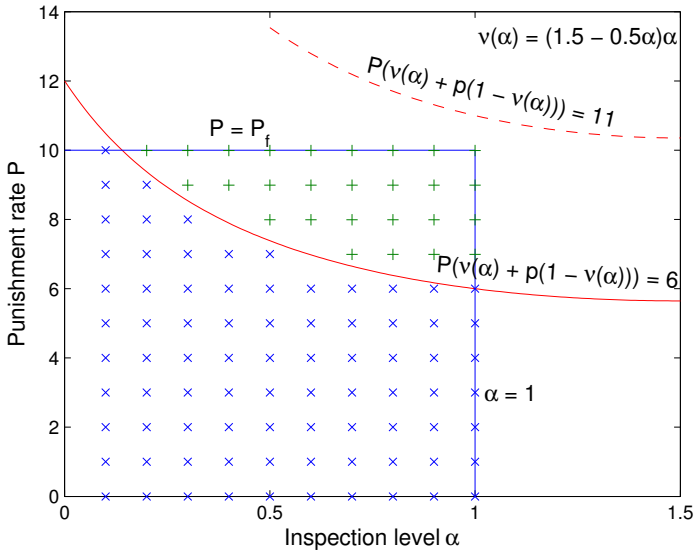

Figure 6.1: Non-deterred (×) and deterred (+) region for $I = \$6$. $I = \$11$ has empty deterred region.

and the utility for the adversary is

$$Iv - Pv(\nu(\alpha) + p(1 - \nu(\alpha)))$$

Graphical representation: A graphical representation of the utilities helps illustrate the ideas presented in the next two sections. (See Figure 6.1). Consider the 2-dimensional plane $R^{\alpha,P}$ spanned by α and P. We define a feasible audit space in $R^{\alpha,P}$ given by $0 \leq \alpha \leq 1$ and $0 \leq P \leq P_f = 10$. We assume $\nu(\alpha) = \mu\alpha - (\mu - 1)\alpha^2$, with efficiency factor $\mu = 1.5$. $\mathcal{D}$'s actions are points in the feasible region. The expected utility of the adversary in each round is given by $v(I - P(\nu(\alpha) + p(1 - \nu(\alpha))))$. Thus, the curve in $R^{\alpha,P}$ given by $I = P(\nu(\alpha) + p(1 - \nu(\alpha)))$ is the separator between positive and negative expected utility regions for the adversary in each round. Within the feasible region, we call the region of positive expected utility the *non-deterred region* and the region of negative utility the *deterred region*.

The equilibrium considered in this model is the myopic optimization presented in the last two chapters on rational adversaries, which results in a Stackelberg equilibrium. However, the interesting scenario is when the myopic optimization is repeated in each round, with the additional consideration of trembling hand of the adversary.

We discuss these in the next chapter.

6.3 Equilibrium in the Audit Game

Our first result is that repeating the same commitment in each round (unaffected by history) followed by the best response of the rational adversary is a SPE. As this result is not surprising or difficult to prove, we prove it informally, relying on the single stage deviation property of SPE. First, as the adversary plays a best response in each round, clearly deviating in one round followed by sticking to its strategy is not beneficial for it (given the defender keeps playing his strategy). Second, given the adversary plays a best response in each round, if the auditor deviates from the per-round Stackelberg commitment then it obtains lower payoff (by definition of Stackelberg equilibrium). Thus, single stage deviation is not beneficial for the auditor also.

The more interesting scenario is with near-rational adversaries. Instead of being perfectly rational, we model the adversary as playing with a *trembling hand* [27]. Whenever the adversary chooses to attack some number of targets v of a given type in given round t, she does so with probability $1-\epsilon_{th}$, but, with (small) probability ϵ_{th} she attacks some other number of target. In other words, we allow the adversary to act completely arbitrarily when she makes a mistake. Then we obtain the following result

Theorem 4. *Repeating the optimal commitment strategy for given targets from last section with the adversary playing the best response with trembling hand probability ϵ_{th}, yields a $(\epsilon_1, 0)$-SPE with near-rational adversaries, where ϵ_1 is*

$$\epsilon_{th} \max_{\alpha} U_{K,\alpha}$$

where $U_{K,\alpha}$ is

$$R - R_{int}K(1-\nu(\alpha)) - R_{ext}K\nu(\alpha) - Cs - aP$$

that is, the expected utility of the defender when the maximum possible number of accesses K are made and all are violations with defender's action α.

Proof. The max. reduction in utility when the adversary attacks some other number of targets than the number v it should have attacked is given by $\max_\alpha U_{K,\alpha}$. Since

this happens only with probability ϵ_{th} the maximum difference in expected payoff for the defender is $\epsilon_{th} \max_\alpha U_{K,\alpha}$. As this bound on difference is same in all the rounds, choosing ϵ_1 as $\epsilon_{th} \max_\alpha U_{K,\alpha}$ satisfies the definition of $(\epsilon_1, 0)$-SPE. The zero approximation for the adversary is obvious, but, due to the near-rationalness the adversary will tremble in his move. □

In addition, we can also consider uncertainty in the knowledge about the adversary. In such a scenario, the defender must make his optimization robust by accounting for the uncertainty in the constraints for the adversary. In our paper [70], we show the effect of this uncertainty is to add to the approximation factor of the SPE. Since that result is quite intuitive and straightforward we skip the description here. In fact, we refer the reader to recent work on uncertainty [71] about a model that handles many nuances of uncertainty in the single shot setting. The approximation in the single shot setting carries over to the approximation factor in the SPE, as shown in the proof above.

6.4 Predictions and Interventions

In this section, we use the model to predict observed practices in industry and the effectiveness of public policy interventions in encouraging organizations to adopt accountable data governance practices (i.e., conduct more thorough audits) by analyzing the equilibrium audit strategy under varying parameters. The explanation of observed practices provides evidence that our audit model is not far from reality.

For this part, we assume all targets (accesses made to EHR) have the same value for the adversary and for the defender, i.e, they are of the same type. We remind the reader of parameters of our model, namely R_{ext} the cost of external detection, R_{int} the cost of external detection, p the probability of external detection and a the factor for cost of setting a punishment level. The predictions are based on the data provided in our paper [70]. In that paper, we used \$10 as the upper bound on x, used $\epsilon_{th} = 0.03$, adversary's utility in committing a violation as \$6 (both when audited or not), the average cost of auditing C was assumed \$50 and $\nu(\alpha) = \mu\alpha - (\mu - 1)\alpha^2$ with $\mu = 1.5$.

Prediction 1: Increasing R_{ext} and p is an effective way to encourage organizations to audit more. We vary R_{ext} from \$5 to \$3900, with R_{int} fixed at \$300. The

R_{ext}	x	α
5	0	0
31	10	0.2
3308	9	0.35
3695	8	0.5

Table 6.1: x, α for varying R_{ext} with $p = 0.5$

equilibrium results are shown in Table 6.1 and Table 6.2 for the two cases: $p = 0.5$ and $p = 0.9$. Graphically, we show the same data in Figure 6.2. In fact, when $p*R_{ext}$ is low organization's may not audit at all. Thus, organizations audits to protect itself from greater loss incurred when violations are caught externally. Surprisingly, organizations may continue to increase inspection level α (incurring higher cost) beyond the minimum level necessary to deter a rational employee. They do so because the employee is not fully rational: even if the employee is deterred there is an ϵ_{th} probability of violations occurring.

Suggested Intervention 1: Subject organizations to external audits and fines when violations are detected. For example, by awarding contracts for conducting 150 external audits by 2012 [72], HHS is moving in the right direction by effectively increasing p. This intervention is having an impact: the 2011 Ponemon study on patient privacy [73] states—"Concerns about the threat of upcoming HHS HIPAA audits and investigation has affected changes in patient data privacy and security programs, according to 55 percent of respondents."

Prediction 2: Interventions that increase the expected loss for both external and internal detection of violations are not as effective in increasing auditing as those that increase expected loss for external detection of violations only. Table 6.3 shows the equilibrium inspection level as R_{ext} and R_{int} are both increased at the same rate. While the inspection level may initially increase, it quickly reaches a peak. As an example, consider the principle of breach detection notification used in many data breach laws [49]. The effect of breach detection notification is to increase both R_{int} and R_{ext} since notification happens for all breaches. While there isn't sufficient data for our model to predict whether these laws are less effective than external audits

R_{ext}	x	α
5	7.5	0
1671	7	0.2
1909	7	0.35
2140	7	0.5
2578	7	0.7
3077	7	0.85
3684	6.5	1.0

Table 6.2: x, α for varying R_{ext} with $p = 0.9$

(see suggested study below), prior empirical analysis [49] indicate that the benefit in breach detection from these laws is only about 6% (after adjusting for increased reporting of breaches due to the law itself).

Suggested study: An empirical study that separately reports costs incurred when violations are internally detected from those that are externally detected would be useful in quantifying and comparing the effectiveness of interventions. Existing studies either do not speak of these distinct categories of costs [11, 49] or hint at the importance of this distinction without reporting numbers [12, 13].

Prediction 3: Employees with higher value for a (e.g., doctors have higher a; suspending a doctor is costlier for the hospital than suspending a nurse) will have lower punishment levels. If punishments were free, i.e., $a = 0$, (an unrealistic assumption) organizations will always keep the punishment rate at maximum according to our model. At higher punishment rates ($a = 1000$), organizations will favor increasing inspections rather than increasing the punishment level (see Table 6.4). While we do not know of an industry-wide study on this topic, there is evidence of such phenomena occurring in hospitals. For example, in 2011 Vermont's Office of Professional Regulation, which licenses nurses, investigated 53 allegations of drug diversion by nurses and disciplined 20. In the same year, the Vermont Board of Medical Practice, which regulates doctors, listed 11 board actions against licensed physicians for a variety of offenses. However, only one doctor had his license revoked while the

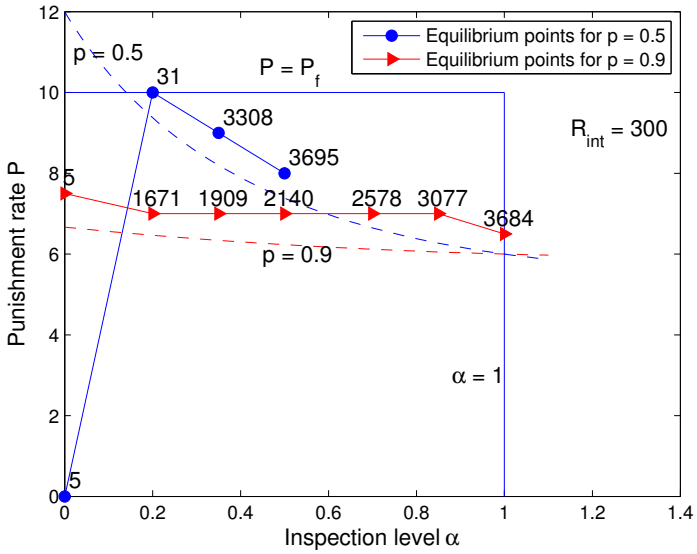

Figure 6.2: Separators for two values of external detection probability p indicated by dashed lines. Equilibrium punishment and inspection rates (P, α) marked on solid lines (see legend) as the reputation loss from external detection R_{ext} varies; the R_{ext} values are labeled above the corresponding equilibrium points.

R_{ext} and R_{int}	x	α
5 to 26	0	0
26 to 3900	10	0.2

Table 6.3: x, α for constant (0) difference in R_{int}, R_{ext}

rest were allowed to continue practicing [4].

Prediction 5: If audit cost C decreases or the efficiency factor μ increases, then the equilibrium inspection level increases. The data supporting this prediction is presented in Table 6.5 and 6.6. Intuitively, it is expected that if the cost of auditing goes down then organizations would audit more, given their fixed budget allocated for auditing. Similarly, a more efficient audit tool with fewer false positives will enable the organization to increase its audit efficiency within the fixed budget. For example, MedAssets claims that Stanford Hospitals and Clinics saved $4 million by using automated tools for auditing [74].

R_{ext}	x	α
5 to 443	0	0
443 to 3900	6.5	1

Table 6.4: x, α for $a = 1000$

C	x	α
10	6.5	1
20	6.5	1
30	7.0	0.85
40	7.5	0.65
50	8.0	0.5
60	9.5	0.25
70	10.0	0.2

Table 6.5: x, α for varying C

6.5 Discussion

As public policy and industry move towards accountability-based privacy governance, the biggest challenge is how to operationalize requirements such as internal enforcement of policies. This chapter provides evidence that the mechanisms we have studied provides a sound basis for informed practical enforcement regimes. We point out a number of studies that can be used to infer data required for the game model, while recognizing the need for more scientific studies with similar goals, and suggest specific studies that can help estimate other parameters. While this chapter captures a number of important practical considerations, it leaves many open problems unanswered, such as what is the best SPE for the defender, or how to handle colluding adversaries? We discuss these and other open problems in the next chapter.

μ	P	α
1.0	10.0	0.3
1.2	9.5	0.35
1.3	9.5	0.35
1.40	9.0	0.45
1.5	9.0	0.45
1.6	8.5	0.5
1.7	8.5	0.5

Table 6.6: P, α for varying μ

Chapter 7

Conclusion and Future Directions

Motivated by the inadequacy of access control in many scenarios and the heuristic approaches to auditing in practice, we embarked on the study of auditing process. Our thesis is as follows:

> Effective audit resource allocation and punishment levels can be efficiently computed by modeling the audit process as a game between a rational auditor and a rational or worst-case auditee.

We presented the following results in support of this thesis

- We proposed the first game model for auditing with worst-case adversaries. The auditor balances between the cost of inspecting potential violations (targets) and the loss from violations committed by the auditee. We identified regret as a notion of optimality in the worst-case adversary audit setting.
- We provided a regret minimizing audit algorithm with regret bound $O\left(\sqrt{\frac{\log N}{T}}\right)$, which is better than known regret bounds in the imperfect (partial) information setting [15].
- We proposed a Stackelberg game model of auditing with rational adversaries that generalizes a standard security game model [21] with a punishment level parameter. The auditee balances between benefit from violations (attacking targets) and loss from punishment when detected. We regard each inspection as a resource with the natural consequence that a resource can inspect only one target. Computing the Stackelberg equilibrium in this model requires solving a non-convex optimization problem. The non-convexity arises due to

the punishment parameter x. The equilibrium computation yields probability values p_i^j: the probability of inspection resource s_j inspecting target t_j. These values can be decomposed into pure actions by using the known Birkhoff-von Neumann theorem.

- We provided novel ways of obtaining FPTAS for particular instances of non-convex optimization problems used in computing the Stackelberg equilibrium. In particular, we provided a reduction of the optimization problem to the problem of finding roots of a polynomial in x. We appeal to known methods of FPTAS for root finding to provide efficient approximation for the optimization problem. We also provided an intuitive FPT algorithm obtained by discretizing x.
- We provided novel ways of improving the running time for the FPT algorithms for some instances. The reduction resulted from the ability to eliminate variables p_i^i and keep only the variables $p_i = \sum_j p_i^j$ in the optimization problem, with increase in the number of constraints of the problem. If the number of new constraints are small, the optimization problem can be solved faster. We identified conditions under which the number of new constraints is small, also showing that these conditions capture many real audit scenarios. We achieved a 5x improvement for instances with 200 targets and 100 inspection resources.

We additionally proposed an asymmetric notion of approximate sub-game perfect equilibrium in games with particular relevance for security. This notions allows the defender to deviate from equilibrium with an ϵ additive factor bound on the difference from optimal utility considering fixed equilibrium play by the adversary. However, it restricts the adversary to not deviate at all. This is important, because deviation by the adversary could result in significant loss for the defender. We also proposed a near-rational model of the adversary, in which the adversary was allowed to play any other action other than his equilibrium action with a small probability.

7.1 Limitations and Future directions

This dissertation takes a first step in the direction of computing optimal resource allocation and punishment levels in the context of auditing. There are several promising directions to explore in models, algorithms and applications.

One direction is to explore variants of our models that reflect constraints that occur in different real world scenarios. A good source for such variants is the literature on security games [21]. As rational audit games generalize security games, many of these variations can be explored for rational audit games also. An example of such variation is considering the cost of resources in the equilibrium computation instead of fixing them apriori, as we do for our worst-case adversary case. Also, it is interesting to explore the possible impacts of punishment, specifically exemplary punishments, i.e., using the actual act of punishment as an example and warning for other potential violators.

Another set of interesting research problems is obtained by relaxing the assumptions in our model. Our model includes assumption such as attack on a target does not affect the security of other targets (no propagation of attack), adversaries do not collude and every target can be audited by a single inspection resource. Relaxing these assumptions results in challenging research problems. An assumption that can be relaxed easily is that violations are caught when inspected; this can be modeled by introducing a probability of catching violations when inspected.

There are a number of open problems in the rational repeated games settings. In our treatment of the rational repeated games setting, we repeat the single round commitment in every round. This approach is limited by the fact that it fails to capture the adversary's adaptiveness in the repeated interaction scenario, thereby providing solution only for a myopic adversary. Another challenging research problem is to explore the dynamics of the equilibrium as we increase the rationality approximation factor ϵ_{th}. The study of adaptiveness in the repeated setting could enable explanation of commonly observed punishment practices, such as *three strikes* policy, i.e., harsh punishment after three violations.

From an algorithmic perspective, some of the challenging problems to explore are as follows: First, while our FPT approach works for the general case of audit games, does there exist a FPTAS for the general case? Second, we show sufficient conditions for polynomially many induced connected sub-graphs of a given graph; can there be a tight(er) characterization of graphs that have polynomially many induced connected sub-graphs? Additionally, some of the variants of the model we suggested above will require algorithmic techniques beyond those we have presented in this dissertation.

For applications, one immediate direction is to optimize for punishment levels

in the application of security games, whenever possible. In applications of security games punishments are considered fixed. However, if the defenders are not bound by law to fix the punishment level, optimizing the punishment level should lead to better utility for the defender. This is especially true for scenarios when the adversaries are motivated by monetary gains, such as train travelers not buying tickets [75]. In the long term, we envision that this work will produce tools that will help organizations to better manage their auditing efforts.

References

[1] G. Hulme. Steady Bleed: State of HealthCare Data Breaches, September 2010. InformationWeek.

[2] *HIPAA Enforcement*, 2010 (accessed November 19,2010). URL http://www.hhs.gov/ocr/privacy/hipaa/enforcement/index.html.

[3] C. Ornstein. Breaches in privacy cost Kaiser, May 2009. Available online at: http://articles.latimes.com/2009/may/15/local/me-privacy15.

[4] K. Picard. Are Drug-Stealing Nurses Punished More Than Doctors?, 2012. URL http://www.7dvt.com/2012are-drug-stealing-nurses-punished-more-doctors.

[5] M. C. Tschantz, A. Datta, and J. M. Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 176–190. IEEE, 2012.

[6] D. Garg, L. Jia, and A. Datta. Policy Monitoring over Evolving Audit Logs: A Logical Method for Privacy Policy Enforcement. Technical report, 2011.

[7] C. A. Gunter, D. M. Liebovitz, and B. Malin. Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE Security & Privacy Magazine*, 9(5), September/October 2011.

[8] A. X. Zheng, B. Liblit, and M. Naik. Statistical debugging: simultaneous identification of multiple bugs. In *Proceedings of the 23rd International Conference on Machine learning*, pages 1105–1112, 2006.

[9] P. Bodik, M. Goldszmidt, A. Fox, D. B. Woodard, and H. Andersen. Fingerprinting the datacenter: automated classification of performance crises. In *Proceedings of the 5th European conference on Computer systems*, EuroSys '10, 2010.

[10] Fairwarning. Industry Best Practices for Patient Privacy in Electronic Health Records, April 2011. URL http://www.fairwarning.com/documents/2011-WHITEPAPER-REMEDIATION-SURVEY.pdf.

[11] Ponemon Institute, LLC. 2010 Annual Study: U.S. Cost of a Data Breach, March 2011. URL http://www.symantec.com/content/en/us/about/media/pdfs/symantec_ponemon_data_breach_costs_report.pdf?om_ext_cid=biz_socmed_twitter_facebook_marketwire_linkedin_2011Mar_worldwide_costofdatabreach.

[12] Verizon. 2012 Data Breach Investigations Report, 2012. URL http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf.

[13] Ponemon Institute, LLC. Benchmark Study on Patient Privacy and Data Security, November 2010. URL http://www.dgshealthlaw.com/uploads/file/Ponemon_Benchmark_Study_on_Patient_Privacy_and_Data_Security%5B1%5D(1).pdf.

[14] Ponemon Institute, LLC. 2011 Cost of Data Breach Study: United States, March 2012. URL http://www.symantec.com/content/en/us/about/media/pdfs/b-ponemon-2011-cost-of-data-breach-us.en-us.pdf?om_ext_cid=biz_socmed_twitter_facebook_marketwire_linkedin_2012Mar_worldwide__CODB_US.

[15] A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pages 79–102, 2007.

[16] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[17] A. Blum and Y. Mansour. From external to internal regret. In *COLT*, pages 621–636, 2005.

[18] J. Blocki, N. Christin, A. Datta, and A. Sinha. Regret minimizing audits: A learning-theoretic basis for privacy protection. In *CSF*, 2011.

[19] Cole Petrochko. DHC: EHR Data Target for Identity Thieves, December 2011. URL http://www.medpagetoday.com/PracticeManagement/InformationTechnology/30074.

[20] American National Standards Institute(ANSI)/The Santa Fe Group/Internet Security Alliance. The financial impact of breached protected health information, accessed May 1,2012. URL http://webstore.ansi.org/phi.

[21] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[22] Shashi Mittal and Andreas S Schulz. An fptas for optimizing a class of low-rank functions over a polytope. *Mathematical Programming*, 141(1-2):103–120, 2013.

[23] J. Blocki, N. Christin, A. Datta, A. D. Procaccia, and A. Sinha. Audit games. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 41–47. AAAI Press, 2013.

[24] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM Conference on Electronic Commerce*, 2006.

[25] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Technical report, University of Tübingen, 1982.

[26] U.S. Supreme Court. Solem v. helm, 463 u.s. 277 (1983), 1983. Available online at: http://caselaw.lp.findlaw.com/scripts/getcase.pl?court=us&vol=463&invol=277.

[27] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.

[28] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton university press, 1944. URL http://books.google.com/books?id=AUDPAAAAMAAJ.

[29] J.F. Nash. *Non-cooperative Games*. Princeton University, 1950. URL http://books.google.cz/books?id=zxtUHAAACAAJ.

[30] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Intelligence Research*, 41(2):297–327, May 2011. ISSN 1076-9757.

[31] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, 2010.

[32] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[33] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956. URL http://projecteuclid.org/euclid.pjm/1103044235.

[34] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957. URL http://ljsavage.wharton.upenn.edu/~steele/Resources/Projects/SequenceProject/Hannan.pdf.

[35] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003. ISSN 0097-5397.

[36] B. W. Lampson. Computer security in the real world. *IEEE Computer*, 37(6): 37–46, 2004.

[37] J. A. Vaughan, L. Jia, K. Mazurak, and S. Zdancewic. Evidence-based audit. In *CSF*, pages 177–191, 2008.

[38] L. Bauer, S. Garriss, and M. K. Reiter. Detecting and resolving policy misconfigurations in access-control systems. In *SACMAT*, pages 185–194, 2008.

[39] J. G. Cederquist, R. Corin, M. A. C. Dekker, S. Etalle, J. I. den Hartog, and G. Lenzini. Audit-based compliance control. *Int. J. Inf. Sec.*, 6(2-3):133–151, 2007.

[40] X. Zhao and M. E. Johnson. Access governance: Flexibility with escalation and audit. In *HICSS*, pages 1–13, 2010.

[41] P. Cheng and P. Rohatgi. IT Security as Risk Management: A Research Perspective. *IBM Research Report*, RC24529, April 2008.

[42] P. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy Multi-Level Security : An Experiment on Quantified Risk-Adaptive Access Control. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2007.

[43] M. D. Vose, A. H. Wright, and J. E. Rowe. Implicit parallelism. In *IN GECCO (2003)*, pages 1505–1517, 2003.

[44] J. Feigenbaum, A. D. Jaggard, and R. N. Wright. Towards a formal model of accountability. In *Proceedings of the 2011 workshop on New security paradigms workshop*, 2011.

[45] NIST. Guide for Conducting Risk Assessments, September 2011. URL http://csrc.nist.gov/publications/drafts/800-30-rev1/SP800-30-Rev1-ipd.pdf.

[46] PricewaterhouseCoopers. A practical guide to risk assessment, December 2008. URL http://www.pwc.com/en_US/us/issues/enterprise-risk-management/assets/risk_assessment_guide.pdf.

[47] K. H. Vellani. Strategic Healthcare Security, Risk Assessments in the Environment of Care, 2008. URL http://www.whea.com/resources/whearisk assessmentarticletoemailafter.pdf. Report for Wisconsin Healthcare Engineering Association.

[48] N. Zhang, W. Yu, X. Fu, and S. K. Das. Towards effective defense against insider attacks: The establishment of defender's reputation. In *IEEE International Conference on Parallel and Distributed Systems*, pages 501–508, 2008. ISBN 978-0-7695-3434-3.

[49] S. Romanosky, D. Hoffman, and A. Acquisti. Empirical analysis of data breach litigation. In *International Conference on Information Systems*, 2011.

[50] Stephen R. Band, Dawn M. Cappelli, Lynn F. Fischer, Andrew P. Moore, Eric D. Shaw, and Randall F. Trzeciak. Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis. Technical Report CMU/SEI-2006-TR-026, Carnegie Mellon University, December 2006.

[51] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. Guards - innovative application of game theory for national airport security. In *IJCAI*, pages 2710–2715, 2011.

[52] J. Pita, M. Jain, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Armor security for los angeles international airport. In *AAAI*, pages 1884–1885, 2008.

[53] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 689–696, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-6-1. URL http://dl.acm.org/citation.cfm?id=

1558013.1558108.

[54] H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Datta. Experiences in the logical specification of the HIPAA and GLBA privacy laws. In *Proceedings of the 9th annual ACM Workshop on Privacy in the Electronic Society (WPES)*, 2010.

[55] US Congress. Health Insurance Portability and Accountability Act of 1996, Privacy Rule. 45 CFR 164, 2002. Available at http://www.access.gpo.gov/nara/cfr/waisidx_07/45cfr164_07.html.

[56] V. Dani and T.P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, page 943. ACM, 2006.

[57] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. *Advances in Neural Information Processing Systems*, 20:1729–1736, 2008.

[58] B. Awerbuch and R. Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008. ISSN 0022-0000.

[59] M. Feldman, C. H. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010–1019, 2006.

[60] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 0162-1459.

[61] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002. ISSN 0022-0000.

[62] G. S. Becker. Crime and punishment: An economic approach. *Journal of Political Economy*, 76:169, 1968. URL http://ideas.repec.org/a/ucp/jpolec/v76y1968p169.html.

[63] H. von Stackelberg. *Marktform und Gleichgewicht. - Wien & Berlin: Springer 1934. VI, 138 S. 8i£¡*. J. Springer, 1934. URL http://books.google.com/

books?id=wihBAAAAIAAJ.

[64] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[65] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *STOC*, 1984.

[66] W. G. Horner. A new method of solving numerical equations of all orders, by continuous approximation. *Philosophical Transactions of the Royal Society of London*, 109:308–335, 1819. doi: 10.1098/rstl.1819.0023. URL http://rstl.royalsocietypublishing.org/content/109/308.short.

[67] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 689–696. International Foundation for Autonomous Agents and Multiagent Systems, 2009. ISBN 978-0-9817381-6-1. URL http://dl.acm.org/citation.cfm?id=1558013.1558108.

[68] G. Birkhoff. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A.*, 5:147–151, 1946.

[69] L. Dulmage and I. Halperin. On a theorem of Frobenius-König and J. von Neumann's game of hide and seek. *Trans. Roy. Soc. Canada. Sect. III. (3)*, 49: 23–29, 1955.

[70] J. Blocki, N. Christin, A. Datta, and A. Sinha. Audit mechanisms for provable risk management and accountable data governance. In *GameSec*, 2012.

[71] T. H. Nguyen, A. X. Jiang, and M. Tambe. Stop the compartmentalization: Unified robust algorithms for handling uncertainties in security games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 317–324, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1. URL http://dl.acm.org/citation.cfm?id=2615731.2615784.

[72] U.S. Department of Health & Human Services. HIPAA Privacy and Security Audit Program. URL http://www.hhs.gov/ocr/privacy/hipaa/enforcement/audit/index.html.

[73] Ponemon Institute, LLC. Second Annual Benchmark Study on Patient Privacy and Data Security, December 2011. URL http://www2.idexpertscorp.com/assets/uploads/PDFs/2011_Ponemon_ID_Experts_Study.pdf.

[74] MedAssets. MedAssets Case Sudy: Stanford hospital takes charge of its charge capture process, increasing net revenue by 4 million, 2011.

[75] Z. Yin, A. X. Jiang, M. P. Johnson, and J. P. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. 2012.

Appendix A: Proofs for Chapter 3

Our main goal is to prove our main theorem from section 3.3.2, showing that our audit mechanism (RMA) achieves low regret with high probability.

First, we prove an orthogonal result in appendix A.1. We prove that our example estimator function from section 3.5 is accurate for the example loss function that we provided.

In appendix A.2 we prove that Lemma 3 holds for *any* estimator function est that satisfies the accuracy and independence properties outlined in section 3.5. So, with high probability the defender's actual regret will be close to his estimated regret.

In appendix A.3 we review standard regret bounds from the literature on regret minimization [15, 17]. We prove that our algorithm achieves low regret with respect to our estimated loss function.

Finally, in appendix A.4 we combine our results to prove our main theorem in section 3.3.2. This theorem shows that, except with probability ϵ, RMA will achieve low regret.

A.1 Estimating Losses

Recall that our regret bounds for algorithm 1 depended on the *accuracy* of the loss function estimator est. We prove that our example estimator (Algorithm 2) from section 3.5 is accurate.

Reminder of Claim 1. *When* $\tilde{\mathbf{L}}^t = \mathsf{est}\left(\mathbf{O}^t_{int}, s^t\right)$ *from algorithm 2*

$$E\left[\mathbf{X}^t_s\right] = 0 \ .$$

Proof. First observe that

$$\begin{aligned} E\left[\tilde{\mathbf{v}}^t\right] &= \frac{1-\beta^{a^t}}{1-\beta^{s^t}} E\left[\mathbf{O}_{int}^t\right] \\ &= \frac{1-\beta^{a^t}}{1-\beta^{s^t}} \sum_{i=1}^{s^t} p_j \\ &= \frac{1-\beta^{a^t}}{1-\beta^{s^t}} \sum_{i=1}^{s^t} v^t \frac{1-\beta}{1-\beta^{a^t}} \beta^{j-1} \\ &= \frac{1-\beta}{1-\beta^{s^t}} v^t \sum_{i=1}^{s^t} \beta^{j-1} \\ &= v^t \ . \end{aligned}$$

From which it follows that

$$\begin{aligned} E\left[\mathbf{X}_s^t\right] &= E\left[\tilde{\mathbf{L}}^t(\vec{s})\right] - E\left[\mathbf{L}^t(\vec{s})\right] \\ &= R - R' \sum_{i=1}^{x} p_j - p \times v^t \times R_{ext} - C \times s - E\left[\mathbf{L}^t(\vec{s})\right] \\ &= R - R' \sum_{i=1}^{x} p_j - p \times v^t \times R_{ext} - C \times s \\ &\quad -R + R' \times E\left[\tilde{\mathbf{v}}^t\right] \times \sum_{j=1}^{x} \left(\frac{1-\beta}{1-\beta^a} \beta^{j-1} \right) + p \times E\left[\tilde{\mathbf{v}}^t\right] \times R_{ext} + C \times s \\ &= R' \times E\left[\tilde{\mathbf{v}}^t\right] \times \sum_{j=1}^{x} \left(\frac{1-\beta}{1-\beta^{a^t}} \beta^{j-1} \right) - R' \sum_{i=1}^{x} p_j \\ &= \left(R' \times \sum_{j=1}^{x} \left(\frac{1-\beta}{1-\beta^{a^t}} \beta^{j-1} v \right) \right) - R' \sum_{i=1}^{x} p_j \\ &= \left(R' \times \sum_{j=1}^{x} p_j \right) - R' \sum_{i=1}^{x} p_j \\ &= 0 \ . \end{aligned}$$

□

A.2 Hoeffding Bounds

Hoeffding Bound [60] bounds the probability of the deviation of a sum of independent random variables from the mean of the sum of random variables. The statement of Hoeffding Bound is as follows: if $X_1, X_2, ..., X_n$ are independent real valued random variables and $a_i \leq X_i \leq b_i$, then for $t > 0$,

$$Pr\left[\left|\sum_{i=1}^{n} X_i - E\left[\sum_{i=1}^{n} X_i\right]\right| > t\right] \leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) .$$

Claim 2. *For every* $\vec{s}$

$$\Pr\left[\left|\mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right)\right| \geq K\right] \leq \frac{\epsilon}{2N} ,$$

where $K = \sqrt{2T \ln \frac{4N}{\epsilon}}$.

Proof. Notice that we can rewrite

$$\mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right) = \sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{X}_{\vec{s}}^{t} .$$

By the independence property of of loss estimator `est`, the random variables $\mathbf{X}_{\vec{s}}^{t}$ are independent. By the accuracy property of our loss function estimator `est` we have

$$E\left[\sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{X}_{\vec{s}}^{t}\right] = 0 .$$

By definition there are exactly $T_{\vec{s}}$ times when $I_{\vec{s}}(t) = 1$ so the sum contains $T_{\vec{s}}$ independent random variables. We also have $-1 \leq \mathbf{X}_{\vec{s}}^{t} \leq 1$ so we can apply Hoeffding Bounds directly to obtain.

$$\Pr\left[\left|\sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{X}_{\vec{s}}^{t}\right| \geq K\right] \leq 2\exp\left(\frac{-2K^2}{2^2 \times T_{\vec{s}}}\right) .$$

Plugging in for K and using the fact that $T_{\vec{s}} \leq T$,

$$\begin{aligned}\Pr\left[\left|\sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{X}_{\vec{s}}^{t}\right| \geq K\right] &\leq 2\exp\left(\frac{-T\ln\frac{4N}{\epsilon}}{T_{\vec{s}}}\right)\\ &\leq 2\exp\left(-\ln\frac{4N}{\epsilon}\right)\\ &= \frac{\epsilon}{2N}\ .\end{aligned}$$

□

Claim 3. *For every* $\vec{s}$

$$\Pr\left[\left|\mathbf{Loss}\left(\mathsf{RMA},\vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA},\vec{s}\right)\right| \geq K\right] \leq \frac{\epsilon}{2N}\ ,$$

where $K = \sqrt{2T\ln\frac{4N}{\epsilon}}$.

Proof. Notice that we can rewrite

$$\mathbf{Loss}\left(\mathsf{RMA},\vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA},\vec{s}\right) = \sum_{t=1}^{T}\sum_{\vec{s}} I_{\vec{s}}(t)p_{\vec{s}}^{t}\mathbf{X}_{\vec{s}}^{t}\ .$$

Set $\mathbf{Y}^t = \sum_{\vec{s}} p_{\vec{s}}^{t}\mathbf{X}_{\vec{s}}^{t}$, and observe that the random variables $\mathbf{Y}^t$ are independent and that $\mathbf{Y}^t \in [-1,1]$. Substituting we get

$$\mathbf{Loss}\left(\mathsf{RMA},\vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA},\vec{s}\right) = \sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{Y}^{t}\ .$$

Applying Hoeffding Bounds we have

$$\Pr\left[\left|\sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{Y}^{t}\right| > K\right] \leq 2\exp\left(\frac{-2K^2}{2^2 T_{\vec{s}}}\right)\ .$$

Set $K = \sqrt{2T \ln \frac{4N}{\epsilon}}$. Then,

$$\begin{aligned}\Pr\left[\left|\mathbf{Loss}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, \vec{s}\right)\right| > \quad K \quad\right] &= \Pr\left[\left|\sum_{t=1}^{T} I_{\vec{s}}(t)\mathbf{Y}^t\right| > K\right] \\ &\leq \quad 2\exp\left(\frac{-2K^2}{2^2 T_{\vec{s}}}\right) \leq 2\exp\left(-\ln\frac{4N}{\epsilon}\right) \\ &\leq \quad \frac{\epsilon}{2N} \ .\end{aligned}$$

□

Lemma 17. *Except with probability ϵ, for all $\vec{s}$ we have*

$$\left|\mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right)\right| < K \ ,$$

and

$$\left|\mathbf{Loss}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, \vec{s}\right)\right| < K \ ,$$

where $K = \sqrt{2T \ln \frac{4N}{\epsilon}}$.

Proof. There are N fixed actions $\vec{s}$ and thus $2N$ total events. Applying the union bound to claims 2 and 3 yields the desired result immediately. □

Claim 4. *Suppose that for all $\vec{s}$ we have*

$$\left|\mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right)\right| < K \ ,$$

and

$$\left|\mathbf{Loss}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, \vec{s}\right)\right| < K \ ,$$

where $K = \sqrt{2T \ln \frac{4N}{\epsilon}}$. then for every $\vec{s}$ we have

$$\mathbf{Regret}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Regret}}\left(\mathsf{RMA}, \vec{s}\right) \leq 2K \ .$$

Proof. We use the definition of $\mathbf{Regret}\left(\mathsf{RMA}, \vec{s}\right)$ and $\widetilde{\mathbf{Regret}}$, and then apply

Lemma 17.

$$\begin{aligned}\mathbf{Regret}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Regret}}\left(\mathsf{RMA}, \vec{s}\right) &= \left(\mathbf{Loss}\left(\mathsf{RMA}, \vec{s}\right) - \mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right)\right) \\ &\quad - \left(\widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right)\right) \\ &\leq \left|\mathbf{Loss}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Loss}}\left(\mathsf{RMA}, \vec{s}\right)\right| \\ &\quad + \left|\mathbf{Loss}\left(\vec{s}, I_{\vec{s}}\right) - \widetilde{\mathbf{Loss}}\left(\vec{s}, I_{\vec{s}}\right)\right| \\ &\leq 2K \ .\end{aligned}$$

□

We are now ready to prove Lemma 3 from section 3.6.

Reminder of Lemma 3.

$$\Pr\left[\exists \vec{s}, \mathbf{Regret}\left(\mathsf{RMA}, \vec{s}\right) - \widetilde{\mathbf{Regret}}\left(\mathsf{RMA}, \vec{s}\right) \geq 2K\right] \leq \epsilon \ ,$$

where $K = \sqrt{2T \ln\left(\frac{4N}{\epsilon}\right)}$.

Proof. We combine claim 4 and Lemma 17. □

A.3 Standard Regret Bounds

We prove upper bounds on our estimated $\widetilde{\mathbf{Regret}}$. The proof techniques used in this section are standard [15, 17]. We include them to be thorough. The following claims will be useful in our proofs.

Claim 5.

$$\sum_{\vec{s} \in \mathsf{AWAKE}^t} w_{\vec{s}}^t \tilde{\mathbf{L}}^t(\vec{s}) = \sum_{\vec{s} \in \mathsf{AWAKE}^t} w_{\vec{s}}^t \tilde{\mathbf{L}}^t(\mathsf{RMA}) \ .$$

Proof. We plug in the definition of $\tilde{\mathbf{L}}^t(\mathsf{RMA})$:

$$\begin{aligned}
\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \tilde{\mathbf{L}}^t(\mathsf{RMA}) &= \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \sum_{\vec{\sigma}} \tilde{\mathbf{L}}^t(\vec{\sigma}) \\
&= \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \sum_{\vec{\sigma}} p_{\vec{\sigma}}^t \tilde{\mathbf{L}}^t(\vec{\sigma}) \\
&= \sum_{\vec{\sigma}} \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t p_{\vec{\sigma}}^t \tilde{\mathbf{L}}^t(\vec{\sigma}) \\
&= \sum_{\vec{\sigma}} \tilde{\mathbf{L}}^t(\vec{\sigma}) \left(\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t p_{\vec{\sigma}}^t \right) \\
&= \sum_{\vec{\sigma}} \tilde{\mathbf{L}}^t(\vec{\sigma}) \left(w_{\vec{\sigma}}^t \right) \\
&\quad \text{Relabel } \vec{\sigma} \\
&= \sum_{\vec{s}} \tilde{\mathbf{L}}^t(\vec{s}) \left(w_{\vec{s}}^t \right) \ .
\end{aligned}$$

□

Claim 6. *For all times t,*

$$\sum_{\vec{s}} w_{\vec{s}}^t \leq N \ .$$

Proof. Initially, $w_{\vec{s}}^0 = 1$ so initially the claim holds,

$$\sum_{\vec{s}} w_{\vec{s}}^0 = N \ .$$

The sum of weights can only decrease. At time t we only update the weights for those experts $\vec{s} \in \mathsf{AWAKE}^t$.

$$\begin{aligned}
\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^{t+1} &= \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \gamma^{\tilde{\mathbf{L}}^t(\vec{s}) - \gamma\tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA})} \\
&= \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \gamma^{\tilde{\mathbf{L}}^t(\vec{s})} \gamma^{-\gamma\tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA})} \\
&\leq \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \Big(1 - (1-\gamma)\,\tilde{\mathbf{L}}^t(\vec{s})\Big) \Big(1 + (1-\gamma)\,\tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA})\Big)
\end{aligned}$$

$$\begin{aligned}
&\leq \left(\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t\right) - (1-\gamma)\left(\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \tilde{\mathbf{L}}^t(\vec{s})\right) \\
&\quad +(1-\gamma)\left(\sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \tilde{\mathbf{L}}^{\mathbf{t}}(\mathsf{RMA})\right) \\
&\quad \text{Apply Claim 5} \\
&\leq \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^t \ ,
\end{aligned}$$

where we used the following two facts

Fact 2.

$$\forall \gamma, y \in [0,1], \gamma^y \leq 1-(1-\gamma)y \ ,$$

and

Fact 3.

$$\forall \gamma, y \in [0,1], \gamma^{-y} \leq 1+(1-\gamma)\frac{y}{\gamma} \ .$$

Therefore,

$$\begin{aligned}
\sum_{\vec{s}} w_{\vec{s}}^{t+1} &= \sum_{\vec{s}\notin\mathsf{AWAKE}^t} w_{\vec{s}}^{t+1} + \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^{t+1} \\
&= \sum_{\vec{s}\notin\mathsf{AWAKE}^t} w_{\vec{s}}^{t} + \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^{t+1} \\
&\leq \sum_{\vec{s}\notin\mathsf{AWAKE}^t} w_{\vec{s}}^{t} + \sum_{\vec{s}\in\mathsf{AWAKE}^t} w_{\vec{s}}^{t} \\
&\leq N \ .
\end{aligned}$$

□

Claim 7. *For each expert $\vec{s}$ we have*

$$\widetilde{\mathbf{Loss}}(\mathsf{RMA}, I_{\vec{s}}) \leq \frac{\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{\ln N}{\ln \frac{1}{\gamma}}}{\gamma} \ ,$$

where N is the total number of experts.

Proof. By assumption, $\tilde{\mathbf{L}}^t(\vec{s})$ is independent of $\vec{s}^t$ so we can think of $\tilde{\mathbf{L}}^t(\vec{s})$ as being fixed before the defender selects its action $\vec{s}^t$. Notice that for all times j we have

$$N \geq w_{\vec{s}}^j = \gamma^{\sum_{t=1}^{j} I_{\vec{s}}(t)\left(\tilde{\mathbf{L}}^t(\vec{s}) - \gamma\tilde{\mathbf{L}}^t(\mathsf{RMA})\right)} .$$

Taking $\log_{\frac{1}{\gamma}}$ we obtain:

$$\begin{aligned} \log_{\frac{1}{\gamma}} N &= \frac{\ln N}{\ln \frac{1}{\gamma}} \\ &\geq -\sum_{t=1}^{j} \left(I_{\vec{s}}(t)\tilde{\mathbf{L}}^t(\vec{s}) - \gamma\tilde{\mathbf{L}}^t(\mathsf{RMA}) \right) \\ &= -\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \gamma\widetilde{\mathbf{Loss}}(\mathsf{RMA}, \mathbf{I}_{\tilde{\mathbf{s}}}) . \end{aligned}$$

Hence,

$$\widetilde{\mathbf{Loss}}(\mathsf{RMA}, I_{\vec{s}}) \leq \frac{\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{\ln N}{\ln \frac{1}{\gamma}}}{\gamma} .$$

□

We are now ready to prove Lemma 1.

Reminder of Lemma 1. *For each expert $\vec{s}$ we have*

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq \frac{1}{L-1}T + 2L\ln N .$$

where N is the total number of experts and γ, our learning parameter has been set to $\gamma = 1 - \frac{1}{L}$.

Proof. Set $\gamma = 1 - \frac{1}{L}$ and apply claim 7. We have

$$\widetilde{\mathbf{Loss}}(\mathsf{RMA}, I_{\vec{s}}) \leq \frac{L}{L-1}\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{L}{L-1}\frac{\ln N}{\ln \frac{L}{L-1}} .$$

Using the definition of $\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s})$ we get

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq \frac{1}{L-1}\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{L}{L-1}\frac{\ln N}{\ln \frac{L}{L-1}} .$$

We will use the following fact

Fact 4.

$$\frac{1}{L-1} < 2\ln\left(\frac{L}{L-1}\right) ,$$

to get

$$\begin{aligned}
\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) &\leq \frac{1}{L-1}\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{L}{L-1}\frac{\log N}{\log \frac{L}{L-1}} \\
&\leq \frac{1}{L-1}\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + \frac{L}{L-1}\frac{\log N}{\frac{1}{2L-2}} \\
&\leq \frac{1}{L-1}\widetilde{\mathbf{Loss}}(\vec{s}, I_{\vec{s}}) + 2L\log N .
\end{aligned}$$

□

Lemma 2 follows immediately from 1.

Reminder of Lemma 2. *For each expert $\vec{s}$ we have*

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq 2\sqrt{2T\ln N} + 2\ln N ,$$

where N is the total number of experts and where our learning parameter has been set to

$$\gamma = 1 - \sqrt{\frac{2\ln N}{T}} .$$

A.4 Main Theorem

We are finally ready to prove our main theorem from section 3.3.2.

Reminder of Theorem 1. *For all $\epsilon > 0$,*

$$\Pr\left[\exists \vec{s},\ \begin{array}{c}\frac{\mathbf{Regret}(\mathsf{RMA},\vec{s})}{T} \geq 2\sqrt{2\frac{\ln N}{T}} + \\ 2\sqrt{\frac{2\ln\left(\frac{4N}{\epsilon}\right)}{T}} + \frac{2}{T}\ln N\end{array}\right] \leq \epsilon .$$

Proof. Lemma 2 tells us that for each expert $\vec{s}$ we have

$$\widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq 2\sqrt{2T \ln N} + 2 \ln N \ .$$

and Lemma 3 tells us that except with probability ϵ, for all actions $\vec{s}$ we have

$$\mathbf{Regret}(\mathsf{RMA}, \vec{s}) - \widetilde{\mathbf{Regret}}(\mathsf{RMA}, \vec{s}) \leq 2\sqrt{2T \ln \left(\frac{4N}{\epsilon}\right)} \ .$$

Combining Lemma 2 with Lemma 3 we obtain the desired result. □

Appendix B: Proofs for Chapter 4

B.5 Missing proofs

Proof of Lemma 4. We prove the contrapositive. Assume there exists a i such that $p_i \neq 0$ and the i^{th} quadratic constraint is not tight. Thus, there exists an $\epsilon > 0$ such that

$$p_i(-x^o - \Delta_i) + p_n^o(x^o + \Delta_n) + \delta_{i,n} + \epsilon = 0 \ .$$

We show that it is possible to increase to p_n^o by a small amount such that all constraints are satisfied, which leads to a higher objective value, proving that p_n^o, x^o is not optimal. Remember that all Δ's are ≥ 0, and $x > 0$.

We do two cases: (1) assume $\forall l \neq n$. $p_n^o(x^o + \Delta_n) + \delta_{l,n} \neq 0$. Then, first, note that p_n^o can be increased by ϵ_i or less and and p_i can be decreased by ϵ_i' to still satisfy the constraint, as long as

$$\epsilon_i'(x^o + \Delta_i) + \epsilon_i(x^o + \Delta_n) \leq \epsilon \ .$$

It is always possible to choose such $\epsilon_i > 0, \epsilon_i' > 0$. Second, note that for those j's for which $p_j = 0$ we get $p_n^o(x^o+\Delta_n)+\delta_{j,n} \leq 0$, and by assumption $p_n^o(x^o+\Delta_n)+\delta_{j,n} \neq 0$, thus, $p_n^o(x^o + \Delta_n) + \delta_{j,n} < 0$. Let ϵ_j be such that $(p_*^o + \epsilon_j)(x^o + \Delta_*) + \delta_{j,*} = 0$, i.e., p_n^o can be increased by ϵ_j or less and the j^{th} constraint will still be satisfied. Third, for those k's for which $p_k \neq 0$, p_n^o can be increased by ϵ_k or less, which must be accompanied with $\epsilon_k' = \frac{x^o+\Delta_*}{x^o+\Delta_i}\epsilon_k$ increase in p_k in order to satisfy the k^{th} quadratic constraint.

Choose feasible ϵ_k''s (which fixes the choice of ϵ_k also) such that $\epsilon_i' - \sum_k \epsilon_k' > 0$.

Then choose an increase in p_i: $\epsilon_i'' < \epsilon_i'$ such that

$$\epsilon_n = \epsilon_i'' - \sum_k \epsilon_k' > 0 \text{ and } \epsilon_n < \min\{\epsilon_i, \min_{p_j=0} \epsilon_j, \min_{p_k \neq 0} \epsilon_k\}$$

Increase p_n^o by ϵ_n, p_k's by ϵ_k' and decrease p_i by ϵ_i'' so that the constraint $\sum_i p_i = 1$ is still satisfied. Also, observe that choosing an increase in p_n^o that is less than any ϵ_k, any ϵ_j, ϵ_i satisfies the quadratic constraints corresponding to p_k's, p_j's and p_i respectively. Then, as $\epsilon_n > 0$ we have shown that p_n^o cannot be optimal.

Next, for the case (2) if $p_n^o(x^o + \Delta_n) + \delta_{l,n} = 0$ for some l then $p_l = 0$, , $\delta_{l,n} < 0$ and the objective becomes

$$p_n \Delta_n - \frac{\delta_{l,n}}{p_n} - \Delta_n \ .$$

Thus, increasing p_n increases the objective. Note that choosing a lower than x^o feasible value for x, results in an higher than p_n^o value for p_n. Also, the k^{th} constraint can be written as $p_k(-x - \Delta_k) + \delta_{k,n} - \delta_{l,n} \leq 0$. We show that it is possible to choose a feasible x lower than x^o. If for some j, $p_j = 0$, then x can be decreased without violating the corresponding constraint. Let p_t's be the probabilities that are non-zero and let the number of such p_t's be T. By assumption there is an $i \neq l$ such that $p_i > 0$ and

$$p_i(-x^o - \Delta_i) + \delta_{i,n} - \delta_{l,n} + \epsilon = 0 \ .$$

For i, it is possible to decrease p_i by ϵ_i such that $\epsilon_i(x^o + \Delta_i) \leq \epsilon/2$, hence the constraint remains satisfied and is still non-tight.

Increase each p_t by ϵ_i/T so that the constraint $\sum_i p_i = 1$ is satisfied. Increasing p_t makes the t^{th} constraint becomes non-tight for sure. Then, all constraints with probabilities greater than 0 are non-tight. For each such constraint it is possible to decrease x (note $x^o > 0$) without violating the constraint.Thus, we obtain a lower feasible x than x^o, hence a higher p_n than p_n^o. Thus, p_n^o, x^o is not optimal. □

Proof of Lemma 6. If $p_n^o(x^o + \Delta_n) + \delta_{i,n} \geq 0$ and $p_n^o(x^o + \Delta_n) + \delta_{j,n} < 0$, where $\delta_{j,n} < \delta_{i,n}$ and $\nexists k$. $\delta_{j,n} < \delta_{k,n} < \delta_{i,n}$, then the exact solution of the i^{th} subproblem will be p_n^o, x^o. Now, since $0 < x \leq 1$ and $0 \leq p_n < 1$, there is one i for which $p_n^o(x^o + \Delta_n) + \delta_{i,n} \geq 0$ and $p_n^o(x^o + \Delta_n) + \delta_{j,n} < 0$, and thus the solution of this sub-problem will return the maximum value. The solution of other sub-problems will return a lower value as the objective is same in all sub-problems. Hence, maximum

of the maximum in each iteration is the global maximum. The approximation case is then an easy extension. □

Proof of Lemma 9.

$$\frac{\Delta_n - \Delta_j}{x - \epsilon + \Delta_j} < \frac{\Delta_n - \Delta_j}{x + \Delta_j} \text{ if } \Delta_n - \Delta_j < 0$$

and using the fact that $\frac{1}{1-\epsilon} < 1 + 2\epsilon$ for $\epsilon < 1/2$,

$$\frac{\Delta_n - \Delta_j}{x - \epsilon + \Delta_j} < \frac{\Delta_n - \Delta_j}{x + \Delta_j} + 2\epsilon \frac{\Delta_n - \Delta_j}{(x + \Delta_j)^2}$$

if $\Delta_n - \Delta_j > 0$ and $\frac{\epsilon}{x+\Delta_j} < 1/2$. The latter condition is true as $\epsilon < B/2$. Thus,

$$\sum_{j:i\le j\le n-1} \frac{\Delta_n - \Delta_j}{x - \epsilon + \Delta_j} < \sum_{j:i\le j\le n-1} \frac{\Delta_n - \Delta_j}{x + \Delta_j} + \sum_{j:i\le j\le n-1, \Delta_n - \Delta_j > 0} \frac{2\epsilon(\Delta_n - \Delta_j)}{(x + \Delta_j)^2}$$

$$\frac{\Delta_n - \Delta_j}{x + \epsilon + \Delta_j} < \frac{\Delta_n - \Delta_j}{x + \Delta_j} \text{ if } \Delta_n - \Delta_j > 0$$

and using the fact that $\frac{1}{1+\epsilon} > 1 - \epsilon$,

$$\frac{\Delta_n - \Delta_j}{x + \epsilon + \Delta_j} < \frac{\Delta_n - \Delta_j}{x + \Delta_j} - \epsilon \frac{\Delta_n - \Delta_j}{(x + \Delta_j)^2}$$

if $\Delta_n - \Delta_j < 0$. Thus,

$$\sum_{j:i\le j\le n-1} \frac{\Delta_n - \Delta_j}{x + \epsilon + \Delta_j} < \sum_{j:i\le j\le n-1} \frac{\Delta_n - \Delta_j}{x + \Delta_j} + \sum_{j:i\le j\le n-1, \Delta_n - \Delta_j < 0} \frac{\epsilon|\Delta_n - \Delta_j|}{(x + \Delta_j)^2}$$

Thus, using the fact that $x > B$ we have

$$\begin{aligned}\sum_{j:i\le j\le n-1}\frac{\Delta_n-\Delta_j}{x_\epsilon+\Delta_j} &< \sum_{j:i\le j\le n-1}\frac{\Delta_n-\Delta_j}{x+\Delta_j}+\\ &\qquad \epsilon\min\left\{\sum_{j:i\le j\le n-1,\Delta_n-\Delta_j<0}\frac{|\Delta_n-\Delta_j|}{(B+\Delta_j)^2},\right.\\ &\qquad \left.\sum_{j:i\le j\le n-1,\Delta_n-\Delta_j>0}\frac{2(\Delta_n-\Delta_j)}{(B+\Delta_j)^2}\right\}\\ &= \sum_{j:i\le j\le n-1}\frac{\Delta_n-\Delta_j}{x+\Delta_j}+\epsilon Y\end{aligned}$$

Very similar to the above proof we also get

$$\begin{aligned}-\sum_{j:i\le j\le n-1}\frac{\delta_{j,n}}{x_\epsilon+\Delta_j} &> -\sum_{j:i\le j\le n-1}\frac{\delta_{j,n}}{x+\Delta_j}-\\ &\qquad \epsilon\min\left\{\sum_{j:i\le j\le n-1,\delta_{j,n}<0}\frac{|\delta_{j,n}|}{(B+\Delta_j)^2},\right.\\ &\qquad \left.\sum_{j:i\le j\le n-1,\delta_{j,n}>0}\frac{2\delta_{j,n}}{(B+\Delta_j)^2}\right\}\\ &= -\sum_{j:i\le j\le n-1}\frac{\delta_{j,n}}{x+\Delta_j}-\epsilon X\end{aligned}$$

Then given

$$p_n=\frac{1-\sum_{j:i\le j\le n-1}\frac{\delta_{(j),n}}{x+\Delta_{(j)}}}{\left(1+\sum_{j:i\le j\le n-1}\frac{x+\Delta_n}{x+\Delta_{(j)}}\right)}=\frac{A}{B}$$

Using the inequalities above

$$F(x_\epsilon)=\frac{1-\sum_{j:i\le j\le n-1}\frac{\delta_{(j),n}}{x_\epsilon+\Delta_{(j)}}}{\left(1+\sum_{j:i\le j\le n-1}\frac{x_\epsilon+\Delta_n}{x_\epsilon+\Delta_{(j)}}\right)}>\frac{A-\epsilon C}{B+\epsilon D}$$

If $p_n \le \psi$, then since $F(x_\epsilon) \ge 0$, we have $F(x) \ge p_n - \psi$. If $p_n \ge \psi$, then since $B > 1$, $A > \psi$ we have

$$\frac{A-\epsilon X}{B+\epsilon Y}>\frac{A}{B}(1-\epsilon(\frac{X}{A}+\frac{Y}{B}))>p_n(1-\epsilon(\frac{X}{\psi}+Y))$$

And, as $p_n < 1$ we have $F(x_\epsilon) > p_n - \epsilon(\frac{X}{\psi} + Y)$. Thus, $F(x_\epsilon) > p_n - \epsilon \min\{\psi, \frac{X}{\psi} + Y\}$. The minimum is less that the positive root of $\psi^2 - Y\psi - X = 0$, that is $\frac{Y+\sqrt{Y^2+4X}}{2}$. □

B.6 Dummy Target

Lemma 18. *If a dummy target t_0 is present for the optimization problem described in Section 4.1, then $p_0 = 0$ at optimal point p_n^o, x^o, where $x^o > 0$.*

Proof. We prove a contradiction. Let $p_0 > 0$ at optimal point. If the problem under consideration is when n represents 0, the objective is not dependent on p_n and thus, then we want to choose x as small as possible to get the maximum objective value. The quadratic inequalities are $p_i(-x^o - \Delta_i) + \delta_{i,0} \leq 0$. Subtracting ϵ from p_0 and adding ϵ/n to all the other p_i, satisfies the $\sum_i p_i = 1$ constraint. But, adding ϵ/n to all the other p_i, allows reducing x^o by a small amount and yet, satisfying each quadratic constraint. But, then we obtain a higher objective value, hence x^o is not optimal.

Next, if the problem under consideration is when n does not represents 0, then an extra constraint is $p_n^o(x^o + \Delta_n) + \delta_{0,n} \leq 0$. Subtracting ϵ from p_0 and adding $\epsilon/(n-1)$ to all the other p_i (except p_n^o), satisfies the $\sum_i p_i = 1$ constraint. Also, each constraint $p_i(-x^o - \Delta_i) + p_n^o(x^o + \Delta_n) + \delta_{i,n} \leq 0$ becomes non-tight (may have been already non-tight) as a result of increasing p_i. Thus, now x^o can be decreased (note $x^o > 0$). Hence, the objective increases, thus p_n^o, x^o is not optimal. □

Appendix C: Proofs for Chapter 5

C.7 Missing Proofs

Proof of Lemma 13. Since the optimization objective depends on the variables p_i's only, we just need to show that the regions spanned by the variables p_i's, as specified by the constraints, are the same in both problems.

First, let p_i's, p_i^j's belong to region given by the grid constraints. Then, by the definition of C, for any c_{ML} we know M is audited only by L. Thus, for $i \in \{t_1, \ldots, t_{|M|}\}$ the variables p_i^j are non-zero only when $j \in \{s_1, \ldots, s_{|L|}\}$. Therefore,

$$\sum_{i \in \{t_1, \ldots, t_{|M|}\}} p_i \leq \sum_{i \in \{t_1, \ldots, t_{|M|}\}} \sum_{j \in \{s_1, \ldots, s_{|L|}\}} p_i^j \leq |L|$$

Hence, p_i's satisfies all constraints in $C \cup \{0 \leq p_i \leq 1\}$, and therefore p_i's belong to region given by $C \cup \{0 \leq p_i \leq 1\}$.

Next, let p_i's belong to region given by $C \cup \{0 \leq p_i \leq 1\}$. We first show that the extreme points of the convex polytope given by $C \cup \{0 \leq p_i \leq 1\}$ sets the variables $p_1, \ldots, p_n$ to either 0 or 1. We show this by contradiction, relying on the fact that extreme points cannot be written as the convex combination of other points in the convex polytope.

Assume, that there exists some values of p_i's for an extreme point z such that $p_k \in (0, 1)$. An extreme point will satisfy some constraints with equality. For any equality constraint, p_k cannot be the only non-integral value, as the constant on the RHS of any constraint is an integer. Thus, for every equality constraint there exists a variable different from p_k that is non-integral. Choose one such variable from each equality constraint, and let this set of variables be P.

Now, consider the point x in which p_k's value is changed to $p_k + \epsilon$, the values of

variables in P are decremented by ϵ and all other variable values remain unchanged. Here ϵ is a very small value, that is smaller than the change that would make $p_k = 1$, or make any variable value in P zero, or violate the inequality constraints. Hence, x lies in the convex polytope. Also, consider the point x' in which the same operations are performed, but, with $-\epsilon$ (and small enough value of ϵ), such that x' also lies in the convex polytope. Thus, it is easy to see that $z = (x + x')/2$. But, that contradicts the assumption that z is a extreme point.

Next, it is easy to see that for given p_i's and p_i^*'s, with corresponding feasible p_i^j's and p_i^{*j}'s, any convex combination $p_i^@$'s of p_i's and p_i^*'s has a feasible solution $p_i^{@j}$'s which is the convex combination of p_i^j's and p_i^{*j}'s. Since, any point in a convex set can be written as the convex combination of its extreme points [], it is enough to show the existence of feasible p_i^j's for the extreme points in order to prove existence of feasible p_i^j's for any point in the convex polytope under consideration.

The extreme points of the given convex polytope has ones in $k' \leq k$ positions and all other zeros. The $k' \leq k$ arises due to the constraint $p_1 + \ldots + p_n \leq k$. Consider the undirected bipartite graph linking the inspections node to the target nodes, with a link indicating that the inspection can audit the linked target. This graph is known from our knowledge of R, and each link in the graph can be labeled by one of the p_i^j variables. Let S' be the set of targets picked by the ones in any extreme points. We claim that there is a perfect matching from S' to the the set of inspection resources (which we prove in next paragraph). Given such a perfect matching, assigning $p_i^j = 1$ for every edge in the matching yields a feasible solution, which completes the proof.

We prove the claim about perfect matching in the last paragraph. We do so by contradiction. Assume there is no perfect matching, then there must be a set $S'' \subseteq S'$, such that $|N(S'')| < |S''|$ (N is neighbors function, this statement holds by the well known Hall's theorem). As $S'' \subseteq S'$ it must hold that $p_i = 1$ for all $i \in S''$. Also, the set of targets S'' is audited only by inspection resources in $N(S'')$ and, by definition of C we must have a constraint (the function index gives the set of indices of the input set of targets)

$$\sum_{i \in index(S'')} p_i \leq |N(S'')| \ .$$

Using, $|N(S'')| < |S''|$, we get

$$\sum_{i \in index(S'')} p_i < |S''| \ .$$

But, since $|index(S'')| = |S''|$, we conclude that all p_i for targets in S'' cannot be one, which is a contradiction. □

C.8 FPT

The quadratic constraint can be rewritten as

$$x(p_n - p_i) - p_i\Delta_i + p_n\Delta_n + \delta_{i,n} \leq 0$$

Suppose we have an optimal point x^o, p_i^o's. Some of the quadratic constraints are tight (say those indexed by k). We decrease x^o by ϵ and decrease p_n by ϵ'. This decrease only affects the quadratic constraints and not the others. Then we need to satisfy

$$(x^o - \epsilon)(p_n^o - p_k^o - \epsilon') - p_k^o\Delta_k + (p_n^0 - \epsilon')\Delta_n + \delta_{k,n} \leq 0$$

Since x^o, p_i^o's are feasible and tight for index k we have

$$x^o(p_n^o - p_k^o) - p_k^o\Delta_k + p_n^0\Delta_n + \delta_{k,n} = 0$$

Thus, we get

$$-\epsilon(p_n^o - p_k^o) - \epsilon' x^o - \epsilon'\Delta_n \leq 0$$

or

$$\epsilon' \geq -\epsilon\frac{p_n^o - p_k^o}{x^o + \Delta_n}$$

Note that the change in objective is $-\Delta_{D,n}\epsilon' + a\epsilon$ which is upper bounded by

$$\epsilon\left(\frac{p_n^o - p_k^o}{x^o + \Delta_n}\Delta_{D,n} + a\right)$$

Since $x > \psi$ and ψ is a constant and the bit precision is also a constant, we get the change in objective to be bounded by $b\epsilon$, where b is a constant. Thus, the

total running time is $O(\mathsf{LP}(n)/\epsilon)$. This argument handles the case when p_n^o is large enough to be reduced by $b\epsilon$.

The case when $p_n^o < b\epsilon$ can be handled by noting that setting $p_n = 0$ and rest of variables at their optimal value is a feasible point. This changes the objective by $\theta(\epsilon)$. Further note that for feasible x^o and $p_n = 0$, all values of $x > x^o$ are also feasible. This can be easily inferred by rewriting the quadratic constraint with $p_n = 0$ as

$$\frac{\delta_{i,n}}{x + \Delta_i} \leq p_i$$

Thus, increasing x by ϵ with $p_n = 0$ still gives a feasible point and further changes the objective by $\theta(\epsilon)$.