Game Theoretic Approaches for Intelligent Auditing

By

Chao Yan

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

January 31, 2022

Nashville, Tennessee

Approved:

Bradley A. Malin, Ph.D.

Yevgeniy Vorobeychik, Ph.D.

Daniel Fabbri, Ph.D.

Murat Kantarcioglu, Ph.D.

Maithilee Kunda, Ph.D.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

This dissertation is the result of a journey that was full of beauty and challenges. I am sincerely grateful to this great period of time because it shaped me to whom I would like to become at the beginning. Reflecting on the past six years, countless memorable moments flash back to my mind. They include the frustration I experienced when I had a difficult time understanding and expressing myself in lab meetings due to language issues, the excitement and sadness when papers were accepted and rejected, the nervousness when I flew toward the empty parking lot alone in dark after concluding the work at lab in the early morning, and the happiness encountered when I was recognized by the community that I love to work in as a researcher. I wish to thank all those who passed along their help and encouragement to me such that this dissertation can be successfully concluded.

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Bradley Malin for his exceptional mentorship, tremendous support, and selfless commitment. Six years ago, received an offer to study for my Ph.D. under Dr. Malin—a change point of my life that brought me here from 8,000 miles away. My training journey departed with clumsiness in almost all aspects of research and communication. Dr. Malin devoted a substantial amount of time to guide me through the process of practicing critical thinking as a scientist, writing good papers, performing high-quality peer review, and developing a good taste of research. Through my training, Dr. Malin largely fostered my intellectual and personal refinement as a young scholar. Dr. Malin also provided me with the freedom and support to work on the research topics of my own interest. His patience and kindness were used to be a shapeless burden that pushed me to grow faster. Over the years, Dr. Malin has been a constant source of inspiration and encouragement whenever I need help. Looking back, I feel extremely fortunate to have this world-class computer and health informatics scientist and the most inspiring and supportive person I have ever seen as my advisor that anyone could ever ask for.

I would also like to thank my committee members: Dr. Yevgeniy Vorobeychik, Dr. Daniel Fabbri, Dr. Murat Kantarcioglu, and Dr. Maithilee Kunda, who have been immensely supportive in providing insightful comments and suggestions to my research. Special thanks to Dr. Vorobeychik, whom I greatly admire from the bottom of my heart for his meticulous scholarship, immense knowledge, sharp thinking, and persistent spirit. I also benefit a lot from his fantastic AI and game theory courses, which strongly motivated me to derive solutions of the real problems presented in this dissertation. I also thank Dr. Fabbri for his generosity of sharing valuable datasets to enable many of my investigations, Dr. Kantarcioglu for his constant support to the research topics I am interested, and Dr. Kunda for her constructive suggestions and remarkable insights to my research.

I have also had the good fortune to collaborate with and also learn from many outstanding researchers, including, but certainly not limited to Dr. You Chen, Dr. Zhijun Yin, Dr. Bo Li (University of Illinois at Urbana-Champaign), Dr. Haifeng Xu (University of Virginia), Dr. Mayur Patel, Dr. Wes Ely, Dr. Jimeng Sun (University of Illinois at Urbana-Champaign), Dr. Tomas Lasko, Dr. David Liebovitz (Northwestern University), Dr. Abel Kho (Northwestern University), Dr. Cheng Gao, Brenda T. Pun, Dr. Wencong Chen, Dr. Prithwish Chakraborty (IBM Watson Health), Dr. Sean Mooney (University of Washington), Yao Yan (University of Washington), Monica Hedda and Hannah Mannering (Loyola University). Particularly, I would like to appreciate Dr. Julia Adler-Milstein (University of California San Francisco), Dr. Jessica Ancker, Dr. Kirk Roberts (University of Texas Health Science Center at Houston) and Dr. Michael F. Chiang (National Eye Institute) for their extraordinary mentorship when I was in the student editorial board of the Journal of the American Medical Informatics Association. I also had the privilege to work with Dr. Xiaoqian Jiang at University of Texas Health Science Center at Houston as a research intern, which left me memorable time in Houston.

Thanks to my wonderful colleagues and alumni from Health Information Privacy Laboratory and neighboring labs who together created fantastic research environment full of

innovation sparks and ideas: Dr. Muqun (Rachel) Li, Dr. Weiyi Xia, Steve Nyemba, Dr. Zhiyu Wan, Dr. Lina Sulieman, Yongtai Liu, Ziqi Zhang, Jia Guo, Thomas Brown, Dr. Wen Zhang, Dr. Wei Xie, Xinmeng Zhang, Congning Ni, Yubo Feng, Eugene Jeong, Victor Borza, Uday Suresh, Dr. Juan Zhao, Dr. Siru Liu, Thomas Li, and Barbara Payne. They always show kindness and give what they can to help me whenever I reach out to them.

In addition, I would like to extend my appreciation to all my friends for being my family here: Diane Jordan, Ellen Zhao, Kevin Zheng, Sixie Yu, Jiayi Yu, Pengfei Wang, Shuang Xie, Liang Tong, Peitung Wei, Tengyu Ma, Xiaojia Li, Xiaoge Zhang, Jing Li, Pumpki Su, Ling Chen, Kathy Zhao, Adam Wang, Yan Sun, Betty Xie, Yue Gao, Tao Zhao, Chengcheng Zhou, Danni Sima, Cai Gao, Yan Chu, Wanqi Chen, Jing Wang, Xizhen Dai, Peng Wang, Xiao Li, Xiaodong Yang, Jian Lou, etc. Special thanks to Dr. Yong Deng and Dr. Qi Liu for their strong support to my career development. Without them, I and my wife would not have been this lucky to meet our advisors and friends at Vanderbilt.

Last but surely not least, I cannot thank my family enough for their unconditional love and complete acceptance of every aspect of me during these years. I am deeply indebted to my wife, Susu Zhang, who is the best gift that completes me and fulfills my life with love and happiness. Over the past decade, my every forward step was seen and accompanied by her, and she shaped my daily life the best I could expect. No amount of words will suffice to express how grateful I am to my parents, who sacrificed much to raise me up, protect my dreams, and give the best they have always. And, finally, thanks to my smart girl cats, Picasso and Koala, for being my stress relievers and happiness amplifiers.

Chao Yan

Vanderbilt University, Nashville, TN

December 2021

TABLE OF CONTENTS

List of Tables

List of Figures

LIST OF ABBREVIATIONS

Electronic health record (EHR)

Healthcare organization (HCO)

Very important person (VIP)

Online Stackelberg signaling policy (OSSP)

Signaling audit game (SAG)

Threat detection and misuse tracking (TDMT)

Vanderbilt University Medical Center (VUMC)

Stackelberg security game (SSG)

Strong Stackelberg Equilibrium (SSE)

Best Response to Quantal Response (BRQR)

Nash Equilibrium (NE)

Optimal auditing problem (OAP)

Column Generation Greedy Search (CGGS)

Iterative Shrink Heuristic Method (ISHM)

Linear programming (LP)

Mixed-integer quadratic program (MIQP)

Part I

Overview and Background

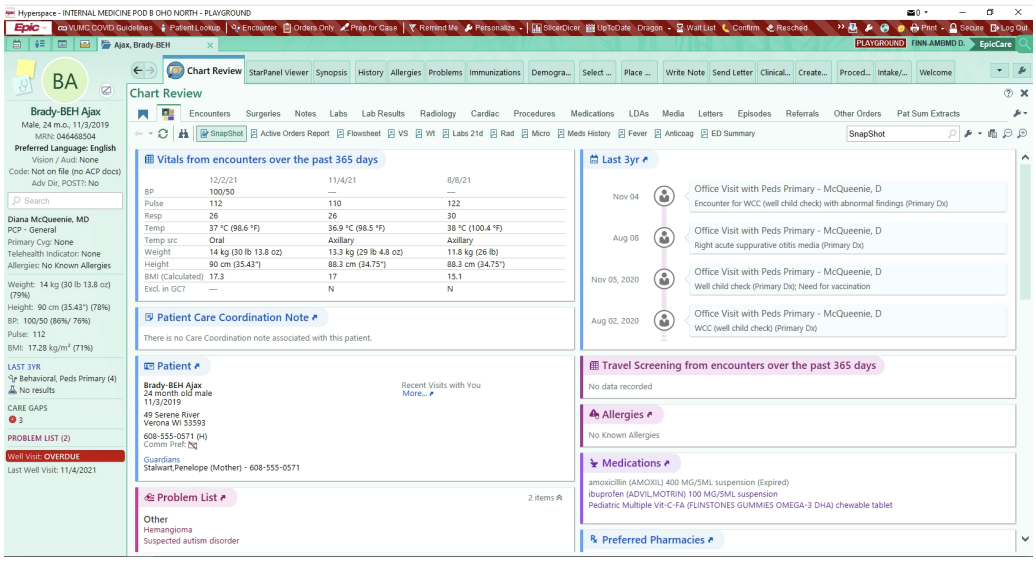
Chapter 1

Overview

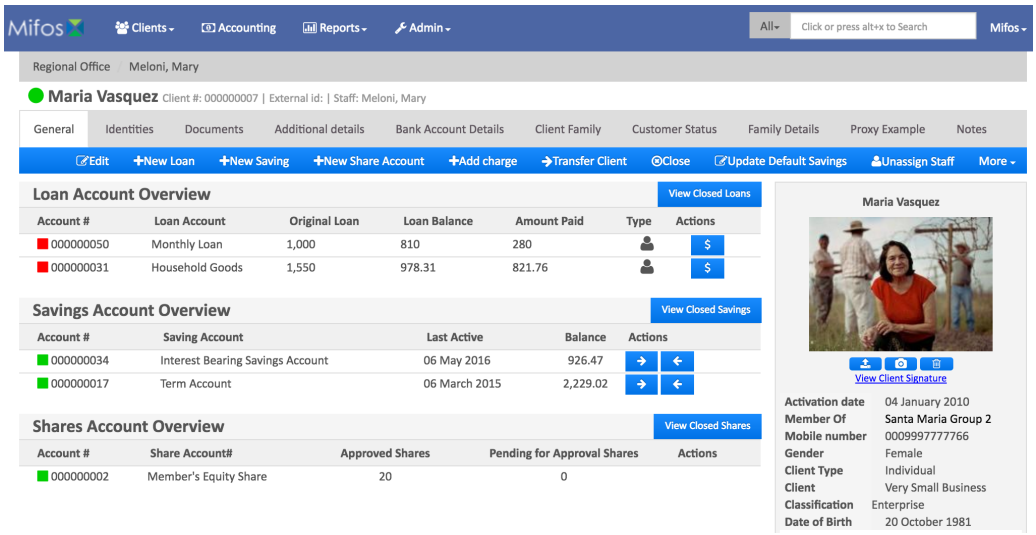
1.1 Introduction

This dissertation asks the question, *"Can we design auditing strategies that are both effective and efficient in defending against data misuse in modern information systems?"*.

The continuous advancement of computation and storage technology has been incentivizing the digitalization of human and our daily life for decades. Such a phenomenon profoundly changes the way information is exchanged, decisions are made, and people think and innovate. With a shared belief of the outstanding capability of improving the efficiency of information exchange and providing assurance to information accuracy and integrity, many modern information systems have emerged to supply critical services to human society by collecting, storing and processing human generated data. An electronic health record (EHR) system is one of these significant innovations (see Figure 1.1a as an example), which enables numerous benefits, including effective communications between clinical personnel and patients [1, 2], care efficiency through anytime access [3], and reductions in medical errors [4, 5]. A financial management information system (see Figure 1.1b as an example) is another remarkable model which enables reliable transaction services, efficient wealth management, and continuous service provision [6]. These systems not only quicken the pace of human activities, but also reshape the nature of daily life.



(a) An examplar interface of *Epic* EHR system showing a fake patient.



(b) An examplar interface of *Mifos* banking system showing a fake client.

Figure 1.1: Concrete domains which motivate and are also directly impacted by the research of this dissertation.

At the same time, attacks are unfortunately never absent due to the important roles that these mission-critical information systems play in facilitating human society, as well as the great value of the data they hold [7, 8, 9]. While attacks can lead to a range of consequences, ranging from the interruption of the continuous operation of information systems to the compromise of data integrity, their final goal often converges to the breach of personal privacy. In 2015, a medical data breach event at *Anthem*—one of the biggest health

insurance providers in the United States—set a new record of data breach in U.S. history [10], affecting over 78.8 million people through a criminal hacking of its data servers. In 2017, the personal identifying data of approximately 145 million of Americans were compromised in an attack against the top credit reporting agency, *Equifax* [11]. Though a large number of manual and automated screening strategies (or combined) guarding security and privacy are continuously developed and deployed, successful attacks against information systems and the sensitive data they hold keep hitting the headlines. As such, it has been widely recognized that no system is impervious to attacks or immune to compromise, especially in the face of the attacks that are adapting, evolving, and improving their ways to undermine protections and to conceal their true purposes.

A widely used solution to defend against data misuse in information systems is to create and then analyze the system audit logs [12, 13, 14, 15]. This simple idea has been practiced quite some time and has leveraged to support multiple goals of information system management [16, 17, 18, 19], including the compliance and accountability in the context of system security and data privacy [20, 21, 22, 23]. Audit logs can be structured heterogenously, but they typically record event details made to a system along the lines of *"who performed which activity at what time point leading to what system status"* [20, 21, 24]. This mechanism is valuable as it enables retrospective investigations for administrators on suspicious events such that real attacks can be recognized and stopped before causing greater losses when being audited. A further step is that, for auditing convenience, suspicious events are usually mapped to predefined semantic types according to their characteristics, each corresponding to a distinct malicious situation [25, 26]. These semantic types can take a variety of forms and excel in screening different threats. For instance, a rule-based mechanism can easily pick out the access activities to the records of very important persons (VIPs) stored in a system, whereas a machine learning detection model can pinpoint the malicious account that demonstrates anomalous system access patterns. The detected suspicious events and their corresponding types are then presented to the system administrator (or auditor)

as *alerts* to be audited, which adds complexity to deriving effective audit policy ahead of time.

However, auditing is non-trivial in practice due to several notable challenges that the auditor can face in the real world domain. First, it is often the case that the audit workload is substantially beyond the available resources for auditing (e.g., the time of security administrators or privacy officials) [27, 28, 29]. Second, a high rate of false positives, resulting from a lack of capability to precisely define maliciousness, makes the auditing inefficient [30, 31, 32]. Third, human attackers usually act strategically according to their knowledge and observations on the system operation to minimize the probability of being caught by the auditor, which makes a fixed auditing pattern vulnerable [33, 34, 35]. For example, an attacker can easily bypass an auditing strategy based on the importance of alert types or a well trained machine learning outlier detection tool by manipulating their attack behaviors. Fourth, compared to the scenario where targets to be protected are fixed as the prior knowledge of both defender and attackers (e.g., airport terminal patrol), the objects to be investigated in data misuse auditing (i.e., alerts) are unknown before an audit cycle (e.g., one day) begins.

Essentially, data misuse auditing is a task that seeks to assign limited investigation resources to a large number of alerts in an adversarial environment. Unfortunately, almost all previous works fail to base their development on this essential characteristic of auditing in deriving their strategies. This dissertation, however, departs from modeling the interactions between an auditor and attackers as a leader-follower game, where an auditor (defender) first commit to a stochastic auditing strategy, and then attackers respond with an attack with certain target or type based on their observations, while attempting to minimize the likelihood of being detected. In fact, the auditing solutions under this modeling architecture demonstrate inherent advantages in comparison to others by incorporating uncertainties to the space via strategy randomization and expanding along realistic incentives of benefit maximization of players. Following this direction of modeling, in this dissertation we ex-

plore the potentiality that a variety of intelligent auditing mechanism designs can achieve to improve the efficiency of the defense and even deterrence against data breach.

1.2 Summary of Contributions

Figure 1.2 summarizes the high level goals and the associated specific game modeling strategies of this dissertation. Basically, this dissertation considers designing audit mechanisms from two different perspectives: *offline prioritization* and *online signaling* (or online warning). Here, we use the terms offline and online to indicate whether there are interactions between the auditor and data users through any auditing mechanism in real time data accessing process. In particular, we unfold our investigations by answering two questions accounting for the adversarial environment between an auditor and attackers: 1) *is it possible to prioritize alerts in an intelligent way such that the auditor can maximize their benefit from this randomized order*, and 2) *can an audit mechanism operate in a real time fashion such that an attacker who is launching attacks can be deterred before success*. The first perspective stems from the observations that in practice system administrators or privacy officials tend to focus on very few alert types that are in their best interest to investigate (or equivalently, the top alert types in their rank of importance). As a consequence, the rest are rarely touched due to budget limitation which creates free lunch for attackers. In addition to conducting auditing totally offline, the second perspective explores incorporating information exchange between players in real time (e.g., when a user request sensitive data) to influence the attackers' strategy selection or even deter attackers. Though our contributions can be applied to general information services, in this dissertation we rely on a representative use case—EHR misuse auditing—to contextualize our investigations, where an employee (or EHR user) of a healthcare organization (HCO) can misuse patients' data and breach patients' privacy via illegal access.

More concretely, to answer the first question (corresponding to Aim 1 in Figure 1.2), we prototype a novel game theoretic audit framework by considering two dimensions si-

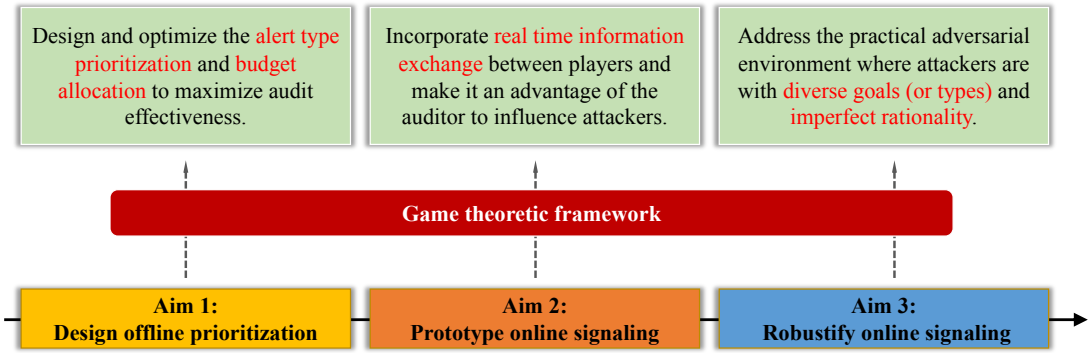


Figure 1.2: The graphic summary of the three main components of this dissertation.

multaneously: 1) how to prioritize the order of types alone which the triggered alerts are investigated retrospectively, and 2) what is the upper bound on how much budget (e.g., human capital or monetary budget) to allocate for auditing each alert type. In this game, the auditor chooses a randomized auditing policy with respect to orders of alert types and a determined budget assignment strategy, while potential attackers choose their record (e.g., EHR) to commit attacks as their responses. We show that even a highly restricted version of the problem is NP-Hard. Nevertheless, we propose a series of algorithmic methods for solving these problems, which leverage a combination of linear programming and column generation to compute a nearly optimal randomized policy for prioritizing alert categories. Using a synthetic dataset on which deriving the exact solution is feasible, we first demonstrate the effectiveness of our methods for approximating the optimal solution with dramatic gains in efficiency. Then, we test the effectiveness of the entire framework with 1) over 1.5 months audit logs of Vanderbilt University Medical Center (VUMC), a major academic medical center in the U.S., where we assign a plausible payoff structure that explicitly represented the gain and loss of players when attackers are caught or not, and 2) a publicly available credit card application dataset. The results of an extensive set of experiments demonstrate that our approach always outperforms the state-of-the-art auditing

strategy (which neglects game theory), regardless of the budget available to the organization. This investigation provides strong evidence that game theory-assisted auditing can favor the auditor by optimizing strategy selection in the adversarial environment. This has been published as a peer-reviewed conference paper [36] and a journal paper [37].

The second research question aims to extend the benefit of adversarial modeling to real time. Specifically, we develop a concept—*online signaling*—and incorporate it into an audit game. At a high level, online signaling functions as follows: whenever a suspicious event starts (e.g., request access to a patient's record, the system configuration files, etc), the system can, in real time, warn the user who made the request (e.g., via a pop-up window with a certain probability to optimize) that *"This Event May Be Audited"*. The user can then choose to stop (if they are an insider and are, thereby, deterred) or proceed with the current action. Then, after a certain period of time, a subset of these events that received signals is audited. Thus, maximizing deterrence via signaling leads us to an online optimization problem, where we must determine 1) whether a warning should be sent and 2) the likelihood that the event will be audited.

As the second research aim of this dissertation (shown in Figure 1.2), we prototype and formalize this auditing problem as a Signaling Audit Game (SAG) as an initial step, where we model the interactions between an auditor and an attacker, and the usability cost (i.e., the phenomenon of deterring normal system users) when being deployed. We term the optimal solution for the auditor as the Online Stackelberg Signaling Policy (OSSP) and prove, in theory, that OSSP is never worse than the optimal solutions achieved in games without signaling. We performed a series of experiments with 10 million EHR access events from VUMC—containing over 26,000 alerts—to illustrate the potential of SAGs and the consistency of their advantages over existing methods. This has been published as a peer-reviewed conference paper [38].

While online signaling-based auditing which leverages the benefit of the auditor's information advantage has the potentiality to outperform the non-signaling strategies, the

SAG can perform poorly in practice due to several critical deficiencies. First, the SAG assumes that all attackers share the same goal such that their preferences over attacking targets are the same. Their preferences are represented by rewards and penalties of both players when an attack is caught or not. In reality, however, the motivations of attackers for compromising a system or sensitive data vary significantly. For example, an employee of an HCO who peeked at a VIP's EHR out of curiosity may be a lesser concern than an employee who sells the same record on a black market (then commit identity theft). Second, following the standard assumption of the security game modeling, the SAG assumes that attackers always act with infallible utility maximizing rationality. However, this is an unreasonably strong assumption because real world attackers may not have the time, energy, or knowledge to perform accurate utility calculations to choose a strategy. And it has been empirically shown that such an assumption in game modeling can cause an excessive loss to the auditor in the face of real world attackers [39] because the auditor can underprotect those targets that they believe an attacker would not likely to attack.

The third aim of this dissertation (shown in Figure 1.2) is to make the online signaling auditing mechanism robust by addressing their deficiencies mentioned above. We introduce a new auditing framework that we call a *robust Bayesian SAG*. First, we model multiple attacker types in the auditing setting by making a Bayesian extension to the SAG, where the uncertainty over the payoffs and preferences of the players are considered by the auditor in selecting their auditing strategies. The resulting problem can then be solved through a compact formulation. Second, to model the imperfect rationality of real world attackers we explore two different types of methods in robust optimization: 1) bounding the worst-case deviation of attackers' strategy selection from their optimal strategy, and 2) constraining the impact of attacker's deviation to the auditor's loss. We incorporate each type of constraints into an algorithm for solving the robust Bayesian SAG in real time and create a corresponding solution concept for each. We investigate the theoretical properties of these solutions and the relationship between them. Surprisingly, these two algorithms, though

with totally different perspectives, can lead to equivalence in certain settings and demonstrate consistency in robustness. To evaluate the performance of the robust Bayesian SAG, we construct two environments: 1) a real environment associated with the audit logs of over 10 million real EHR accesses of VUMC (the same evaluation dataset as in Aim 2) and 2) a simulated controlled environment derived from the real data, which allows us to simulate attacker behaviors regarding their rationality degree. We specifically evaluate the expected utility of the auditor between the our solutions and the state-of-the-art auditing method in different conditions to demonstrate the value of the new auditing solutions and their scalability. This has been submitted to a conference for review.

1.3 Dissertation Structure

The remainder of this dissertation is organized as follows. Chapter 2 surveys the related work. Afterwards, we expand on each of the aforementioned aim by formalizing the corresponding problem to a specific game theoretic model, deriving their solutions and then making evaluations using real and simulated datasets. Specifically, in Chapter 3, we formalize our alert prioritization game and derive its solving algorithms to improve the offline data misuse auditing. In Chapter 4, the concept of online signaling, as well as the resulting model—SAG are introduced, followed by the solution's theoretical properties and performance evaluations. Chapter 5 then proposes the robust framework of the SAG considering multiple attacker types and their imperfect rationality in selecting strategies. We conclude the dissertation in Chapter 6 by summarizing our contribution and discussing the future work.

Chapter 2

Related Work

Auditing has been widely investigated for the protection of critical resources from attacks [40, 41, 42, 43]. There have been a number of investigations into effective alert management strategies and efficient auditing mechanisms for information systems. In this chapter, we review recent developments that are closely related to our investigation.

2.1 Alert Prioritazation

2.1.1 Alert Frameworks

Generally speaking, there are two main categories by which alerts are generated by a system: 1) machine learning methods [44] – which usually measure the distance from either normal or suspicious patterns [45, 46, 47, 48, 49, 50], and 2) rule-based approaches—which flag the occurrences of predefined events when they are observed [51, 52, 53]. Concrete implementations are often tailored to distinct application domains.

In the healthcare sector, methods have been proposed to find misuse of EHR systems. Boxwala et at. [54] treated it as a two-label classification problem and trained support vector machines and logistic regression models to detect suspicious accesses. Given that not all suspicious accesses follow a pattern, various techniques have been developed to determine the extent to which an EHR user [55] or their specific access [56] deviated from the typical collaborative behavior. By contrast, Fabbri et al. [57, 58, 59] designed an explanation-based auditing mechanism which generates and learns typical access patterns from an expert-, as well as data-driven, view. EHR access events by authenticated employees can be explained away by logical relations (e.g., a patient scheduled an appointment with a physician), while the residual can trigger alerts according to predefined rules (e.g., co-workers) or simply fail to have an explanation. The remaining events are provided to

privacy officials for investigation; however, in practice, only a tiny fraction can feasibly be audited due to the resource limitation.

In the financial sector, fraud detection [60] in credit card applications assists banks in mitigating their losses and protecting consumers [61]. Several machine learning-based [62] models have been developed to detect fraud behavior. Some of the notable models include hidden Markov models [63], neural networks [64], support vector machines [65], etc. Rule-based techniques were also integrated into some detection frameworks [66, 67, 68, 69]. While these methods trigger alerts for investigators, they result in a significant number of false positives—a problem which can be mitigated through alert prioritization schemes.

2.1.2 Alert Burden Reduction

Alert prioritization, as a set of methods to make the large number of alerts raised more manageable, is not new in the domain of security and privacy. Various methods have been developed to reduce alert magnitude generated in information systems [70]. Many focus on reducing redundancy and clustering alerts based on their similarity [71, 70, 72, 73, 74]. In particular, a cooperative module was proposed for intrusion detection, which implemented the functions of alert management, clustering and correlation [75]. Xiao et al. proposed a multi-level alert fusion model to abstract high-level attack scenarios to reduce redundancy [76]. As an alternative, fuzzy set theory was applied by Maggi et al. to design robust alert aggregation algorithms [77]. Also, a fuzzy-logic engine to prioritize alerts was introduced by Alsubhi et al. by rescoring alerts based on a few metrics that dynamically characterize the degree of maliciousness of an attacker [78, 79]. Njogu et al. built a robust alert cluster by evaluating the similarity between alerts to improve the quality of those sent to analysts [80]. By contrast, Aminanto et al. [81, 82, 83] and Sun et al. [84] leveraged the temporal- or real time-based analysis to inform an isolation forest model to identify real threats as outliers. However, none of these approaches consider the impact of alert aggregation and prioritization on decisions by potential attackers, especially as the latter may choose attacks

that circumvent the prioritization and aggregation mechanisms.

2.2 Stackelberg Security Games

The developed frameworks in this dissertation are related to the literature on *Stackelberg security game* (SSG) [85, 86, 87], a leader-follower game that well characterizes a variety of adversarial environments where defender and attacker interact with each other under specific security resource constraints. Under this specific category of games, a single or multiple defenders [88, 89] first commit to a (possibly randomized) allocation of security resources, while the attacker chooses an attack in response based on observation or surveillance. The allocation of security resources associates with the selection of strategies for players. Deriving an optimal defensive strategy needs to consider the priority differences regarding the targets to be protected, the possible response of attackers based on their knowledge, and the uncertainties ranging from capabilities and knowledge of players to their types [86]. The most commonly adopted solution concept of a basic SSG is called *Strong Stackelberg Equilibrium* (SSE) [90]. An SSE corresponds to a pair of strategies (one for the defender, and the other for the attacker) such that 1) changing to any other strategies will never lead to an increase of the expected utility of a player, and 2) the attacker breaks ties in favor of the defender. The tie-breaking assumption is mathematically reasonable because the defender is able to induce the favorable strong equilibrium by shifting the protection by an arbitrary small level from the equilibrium such that the attacker strictly prefer the desired strategy [91].

Due to the excellent role the SSG plays in the complex decision making process, such models have been applied in a broad variety of security settings, such as airport checkpoint randomization [92] and passenger screening [93], air marshal scheduling [94], coast guard patrol scheduling [95], and even for preventing poaching and illegal fishing [96]. However, models used in many of these prior works are specialized to physical security and do not readily generalize to the problem of prioritizing alerts for auditing. In practical alert

prioritization and auditing problems, in contrast, a crucial consideration is that there are many potential attackers and many potential victims or modes of attack for each of these. Moreover, auditing policies involve recourse actions where the specific alerts audited depend on the realizations of alerts of various types. Since alert realizations are stochastic, this engenders complex interactions between the defender and attackers, and results in a highly complex space of prioritization policies for the defender.

2.3 Audit Games and Alert Prioritization

Blocki *et al.* first modeled the audit problem between an auditor and an attacker as a classic security game [97, 98]. In this setting, players act strategically and the goal is to learn an optimal resource allocation strategy that optimizes the expected payoff of the auditor [98]. Different from the basic SSG, the auditor employs a continuous punishment rate parameter to deter attackers, which, in turn, influences the auditor's utility due to its impact to organization productivity. This component can be regarded as the usability cost of deploying the audit game. To simulate the real audit environment, Blocki *et al.* generalized the framework by accounting for the situations with multiple defender resources [99]. However, their methods treat alerts as a set of existing targets that could be attacked, a modeling decision that cannot be readily generalized into the system audit setting. To solve this challenge, Schlenker *et al.* introduced a game theoretic approach to deal with how to assign alerts to security analysts [100], where each analyst has different areas of expertise [101]. However, there are two key limitations that hinder its application to the general data misuse auditing: 1) it considers only single attacker, whereas auditing decisions in the context of access control policies commonly involve many potential attackers, with most never considering the possibility of an attack, and 2) it assumes that the number of alerts in each category is known a priori to both the auditor and attacker. In this dissertation, we consider the scenario with multiple attackers and apply the practical situation where alert counts by category are stochastic and can exhibit high variance.

Laszka *et al.* first modeled the alert prioritization problem as a game, in which the auditor determines the order of auditing with respect to alert types [102]. However, this game has two obvious deficiencies by assuming that 1) the identity of a specific attacker was unknown and 2) an exhaustive auditing strategy across alert types of a given order would be applied. These assumptions are relaxed in the investigation addressed by our study in this dissertation. With a similar goal of prioritizing alerts, Tong *et al.* considered the dynamics of detection environment as an important factor and proposed a robust approach that combines game theory and adversarial reinforcement learning [103]. Specifically, neural reinforcement learning was used to compute approximately optimal policies for each player in response to a fixed policy of the other player. Then, the final policies of players can be derived by a double-oracle framework. Though demonstrating improved auditing performance over previous research in controlled testing scenarios, it is unclear whether the strong assumptions on attackers (e.g., a full knowledge of the state of the detection environment) can weaken the protection effectiveness of this model in practice.

2.4 Signaling in security games

It was recently shown that the traditional Stackelberg game framework has limited efficacy in many real world settings, which can be improved by a signaling scheme to reveal noisy information to the attacker [104, 105, 106]. In particular, Xu *et al.* proposed a two-stage security game model to protect targets with a better performance. In the first stage, the defender allocates inspection resources and the attacker selects a target. In the second stage, the defender reveals information, potentially deterring the attacker's attack plan of attack [107]. The advantages of signaling were subsequently extended to Bayesian Stackelberg games, where players have payoff-relevant private information [108]. It has been shown that signaling also boosts defensive performance in security games, specifically for the task of assigning randomized human patrollers and sensors to protect important targets [106]. However, these investigations aimed to protect existing physical targets as well. The

methodology does not easily fit into the auditing environment, where the timing of budget assignment and signaling are reversed.

2.5 Imperfect Rationality

To account for the imperfect rationality of real world attackers, various investigations have worked to integrate the human decision making process into formal algorithms. Based on quantal response, Yang *et al.* developed *Best Response to Quantal Response* (BRQR) to explicitly model the probability that an attacker selects each attack based on their expected utility on each response strategy [109]. Nguyen *et al.* improved the performance by integrating a subjective utility function into BRQR [110], which they named SU-BRQR. Though effective, these models try to optimize the defender's utility with an objective function that is non-linear and non-convex. As a result, time needed to solve the problem leads to a solution that does not scale for real time auditing. Moreover, the proposed solutions required an accurate estimation of the parameter that determines the noise in the human response function, which is non-trivial given the excessively low rate of adversarial events.

Part II

Offline Auditing: Alert Prioritization

Chapter 3

Game Theoretic Alert Prioritization

The quantity of personal data that is collected, stored, and subsequently processed continues to grow rapidly. Given its sensitivity, ensuring privacy protections has become a necessary component of database management. To enhance protection, a number of mechanisms have been developed, such as audit logging and alert triggers, which notify administrators about suspicious activities. However, this approach is limited. First, the volume of alerts is often substantially greater than the auditing capabilities of organizations. Second, strategic attackers can attempt to disguise their actions or carefully choose targets, thus hide illicit activities. In this chapter, we introduce an auditing approach that accounts for adversarial behavior by (1) prioritizing the order in which types of alerts are investigated and (2) providing an upper bound on how much resource to allocate for each type.

Specifically, we model the interaction between a database auditor and attackers as a Stackelberg game. We show that even a highly constrained version of such problem is NP-Hard. Then we introduce a method that combines linear programming, column generation and heuristic searching to derive an auditing policy. On the synthetic data, we perform an extensive evaluation on the approximation degree of our solution with the optimal one. The two real datasets, (1) 1.5 months of audit logs from Vanderbilt University Medical Center and (2) a publicly available credit card application dataset, are used to test the policy-searching performance. The findings demonstrate the effectiveness of the proposed methods for searching the audit strategies, and our general approach significantly outperforms non-game-theoretic baselines.

3.1 Introduction

Modern computing and storage technology has made it possible to create *ad hoc* database systems with the ability to collect, store, and process extremely detailed information about the daily activities of individuals [111]. These database systems hold great value for society, but accordingly face challenges to security and, ultimately, personal privacy. The sensitive nature of the data stored in such systems attracts malicious attackers who can gain value by disrupting them in various ways (e.g., stealing sensitive information, commandeering computational resources, committing financial fraud, and simply shutting the system down) [112]. It is evident that the severity and frequency of attack events continue to grow. Notably, the most recent breach at Equifax led to the exposure of data on 143 million Americans, including credit card numbers, Social Security numbers, and other information that could be used for identity theft or other illicit purposes [11]. Even more of a concern is that the exploit of the system continued for at least two months before it was discovered.

While complex access control systems have been developed for database management, it has been recognized that in practice no database systems will be impervious to attack [113]. As such, prospective technical protections need to be complemented by retrospective auditing mechanisms, a notion that has been well recognized by the database community [114]. Though audits do not directly prevent attacks in their own right, they may allow for the discovery of breaches that can be followed up on before they escalate to full blown exploits by adversaries originating from beyond, as well as within, an organization.

In the general situation of database management, auditing relies heavily on the performance of a *threat detection and misuse tracking* (TDMT) module, which raises real-time alerts based on the actions committed to a system for further investigation by experts. In general, the alert types are specifically predefined by the administrator officials in *ad hoc* applications. For instance, in the healthcare domain, organizations are increasingly reliant on electronic health record (EHR) systems for anytime, anywhere access to a patient's

health status. Given the complex and dynamic nature of healthcare, these organizations often grant employees broad access privileges, which increases the potential risk that inside employees illegally exploit the EHR of patients [115]. To detect when a specific access to a patient's health record is a potential policy violation, healthcare organizations use various triggers to generate alerts, which can be based on predefined rules (e.g., when an access is made to a designated very important person). As a consequence, the detected anomalies, which indicate deviations from routine behavior (e.g., when a pediatrician accesses the records of elderly individuals), can be checked by privacy officials [116]. As another example, consider the credit card provisioning domain. In this setting, individuals are interested in applying for credit cards, which might be used in a fraudulent manner. There may be many reasons why an application would trigger an alert for a credit risk analyst, who, in turn, would need to determine if the applicant is worth investigating.

Although TDMTs are widely deployed in database systems as both detection and deterrence tools, security and privacy have not been sufficiently guaranteed. The utility of TDMT in practice is currently limited by the fact that they often lead to a very large number of alerts, whereas the number of actual violations tends to be quite small. This is particularly problematic because the large quantities of false alarms can easily overwhelm the capacity of the administrative officials who are expected to follow-up on these, but have limited resources at their disposal [117]. One typical example is the observation from our evaluation dataset: at Vanderbilt University Medical Center, on any single workday, the volume of accesses to the EHR system is around 1.8 million, of which more than 30,000 alerts of varying predefined types are generated, which far beyond the capacity of privacy officials. Therefore, in lieu of an efficient audit functionality in the database systems, TDMTs are not optimized for detecting suspicious behavior.

Given the overwhelming number of alerts in comparison to available auditing resource and the need to catch attackers, the core query function invoked by an administrator must consider resource constraints. And, given such constraints, we must determine which trig-

gered alerts should be recommended for investigation. One intuitive way to proceed is to prioritize alert categories based on the potential impact of a violation if one were to be found. However, this is an inadequate strategy because would-be violators can be strategic and, thus, reason about the specific violations they can perform so that they balance the chance of being audited with the benefits of the violation. To address this challenge, we introduce a model based on a Stackelberg game, in which an auditor chooses a randomized auditing policy, while potential violators choose their victims (such as which health records to view) or to refrain from malicious behavior after observing the auditing policy.

Specifically, our model restricts the space of audit policies to consider two dimensions: 1) how to prioritize alert categories and 2) how many resources to allocate to each category. We show that even a highly restricted version of the auditor's problem is NP-Hard. Nevertheless, we propose a series of algorithmic methods for solving these problems, leveraging a combination of linear programming and column generation to compute an optimal randomized policy for prioritizing alert categories. We perform an extensive experimental evaluation with two real datasets—one involving EHR access alerts and the other pertaining to credit card eligibility decisions—the results of which demonstrate the effectiveness of our approach.

The remainder of the chapter is organized as follows. In subsection 3.2, we formally define the game theoretic alert prioritization problem and prove its NP-hardness. In subsection 3.3, we describe the algorithmic approaches for computing a randomized audit policy. In subsection 3.4, we introduce a synthetic dataset to show, in a controlled manner, the effectiveness of our methods for approximating the optimal solution with dramatic gains in efficiency. In subsection 3.5, we use two real datasets (from healthcare and finance) that rely upon predefined alert types to show that our methods lead to high-quality audit strategies. We discuss our findings and conclude this chapter in subection 3.6 and subection 3.7.

3.2 Game Theoretic Model of Alert Prioritization

In environments dealing with sensitive data or critical services, it is common to deploy TDMTs to raise alerts upon observing suspicious events. By defining *ad hoc* alert types, each suspicious event can be marked with an alert label, or type, and put into an audit bin corresponding to this type. Typically, the vast majority of the raised alerts do not correspond to actual attacks, as they are generated as a part of a routine workflow that is too complex to accurately capture. Consequently, looking for actual violations amounts to looking for needles in a large haystack of alerts, and inspecting all, or even a large proportion of, alerts that are typically generated is rarely feasible. A crucial consideration, therefore, is how to *prioritize* alerts, choosing a subset that can be audited given a specified auditing budget from a vast pool of possibilities. The prioritization problem is complicated by the fact that intelligent adversaries—that is, would-be violators of organizational access policies—would react to an auditing policy by changing their behavior to balance the gains from violations, and the likelihood, and consequences, of detection.

We proceed to describe a formal model of alert prioritization as a game between an auditor, who chooses an alert prioritization policy, and multiple attackers, who determine the nature of violations, or are deterred from one, in response. In the described scenarios, we assume that the attackers have complete information, which is the worst case assumption[1]. For reference purposes, the symbols used throughout this chapter are described in Table 3.1.

3.2.1 System Model

Let $\mathcal{E}$ be the set of potential adversaries, such as employees in a healthcare organization, some of whom could be potential violators of privacy policies, and $\mathcal{V}$ be the set of potential victims, such as patients in a healthcare facility. We define events, as well as attacks,

[1] We do not claim that the attacker actually has such information, but instead aim to be robust even if the attacker has complete information

Table 3.1: A legend of the notation used in this chapter.

Symbols	Interpretation
$\mathcal{T}$	Set of alert types
$\mathcal{E}$	Set of entities or users causing events
$\mathcal{V}$	Set of records or files available for access
$P^t(\langle e,v\rangle)$	Probability of raising type t alert by attack $\langle e,v\rangle$
C_t	Cost for auditing an alert of type t
B	Auditing budget
$F_t(n)$	Probability that at most n alerts are in type t
$\boldsymbol{O}$	Set of all alert prioritizations over T
Z_t	Number of alerts under type t
b_t	Budget threshold assigned for auditing type t
$R(\langle e,v\rangle)$	Adversary's gain when attack $\langle e,v\rangle$ is undetected
$M(\langle e,v\rangle)$	Adversary's penalty when attack $\langle e,v\rangle$ is captured
$K(\langle e,v\rangle)$	Cost of deploying attack $\langle e,v\rangle$
$p_{\boldsymbol{o}}$	Probability of choosing an alert prioritization $\boldsymbol{o}$
P_e	Probability that e is a potential adversary

by a tuple $\langle e,v\rangle$. A subset of these events will trigger alerts. Now, let $\mathcal{T}$ be the set of alert types or categorical labels assigned to different kinds of suspicious behavior. For example, a doctor viewing a record for a patient not assigned to them and a nurse viewing the EHR for another nurse (who is also a patient) in the same healthcare facility could trigger two distinct alert types. We assume that each event $\langle e,v\rangle$ maps to at most one alert type $t \in \mathcal{T}$. This mapping may be stochastic; that is, given an event $\langle e,v\rangle$, an alert with type t is triggered with probability $P^t(\langle e,v\rangle)$, and no alert is triggered otherwise (i.e., $p^{t'}_{\langle e,v\rangle} = 0$ for all $t' \neq t$). Typically, both categorization of alerts and corresponding mapping between events and types is given (for example, through predefined rules). If not, it can be inferred by generating possible attacks and inspecting how they are categorized by TDMT. Auditing each alert is time-consuming and the time to audit an alert can vary by alert type. Let C_t be the cost (e.g., time) of auditing a single alert of type t and let B be the total budget allocated for auditing.

We assume that the number of alerts triggered by normal events follows a distribution which reflects a typical workflow of the organization and can be learned based on historical

data. We assume this distribution is known, represented by $F_t(n)$, which is the probability that at most n alerts of type t are generated. If we make the reasonable assumption that attacks are rare events and that the alert logs are tamper-proof by applying a certain technique [118], then this distribution can be obtained from historical alert logs. It is noteworthy that the probability that adversaries successfully manipulate the distribution in the sensitive practices (e.g., the EHR system or the credit card application program), to fool the audit model is almost zero. The cost of orchestrating and implementing such attacks is much higher than what could be gained from running a few undetected attacks.

3.2.2 Game Model

We model the interaction between the auditor and potential violators as a Stackelberg game. Informally, the auditor chooses a possibly randomized auditing policy, which is observed by the prospective violators, who in response choose the nature of the attack, if any. Both decisions are made *before the alerts produced through normal workflow are generated* according to a known stochastic process $F_t(n)$.

In general, a specific pure strategy of the defender (auditor) is a mapping from an arbitrary realization of alert counts of all types to a subset of alerts that are to be inspected, abiding by a constraint on the total amount of budget B allocated for auditing alerts. Even representing a single such strategy is intractable, let alone optimizing in the space of randomizations over these. We, therefore, restrict the defender strategy space in two ways. First, we let pure strategies involve an ordering $\boldsymbol{o} = (o_1, o_2, \ldots, o_{|\mathcal{T}|})$ $(\forall i, j \in \mathbb{Z}^+$ and $i, j \in [1, |\mathcal{T}|]$, if $i \neq j$, then $o_i \neq o_j)$ over alert types, where the subscript indicates the position in the ordering, and a vector of thresholds $\mathbf{b} = (b_1, \ldots, b_{|\mathcal{T}|})$, with b_t being the maximum budget available for auditing alerts in category t. Let $\mathbf{O}$ be the set of feasible orderings, which may be a subset of all possible orders over types (e.g., the organizational policy may impose constraints, such as always prioritizing some alert categories over others). We interpret a threshold b_t as the maximum budget allocated to t; thus, the most alerts

of type t that can be inspected is $\lfloor b_t/C_t \rfloor$. Second, we allow the auditor to choose a randomized policy over alert orderings, with $p_{\boldsymbol{o}}$ being the probability that ordering $\boldsymbol{o}$ over alert types is chosen, whereas the thresholds $\mathbf{b}$ are deterministic and independent of the chosen alert priorities.

We have a collection of potential adversaries $\mathcal{E}$, each of whom may target any potential victim $v \in \mathcal{V}$. We assume that the adversary will target exactly one victim (or at most one, if $\mathcal{V}$ contains an option of not attacking anyone). Thus, the strategy space of each adversary e is $\mathcal{V}$. In addition, we characterize the probability that an adversary $e \in \mathcal{E}$ performs an attack as P_e (i.e., e does not even consider attacking with probability $1 - P_e$).

Suppose we fix a prioritization $\boldsymbol{o}$ and thresholds $\mathbf{b}$. Let $o(t)$ be the position of alert type t in $\boldsymbol{o}$ and o_i be the alert type in position i in the order. Let $B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})$ be the budget remaining to inspect alerts of type t if the order is $\boldsymbol{o}$, the defender uses alert type thresholds $\mathbf{b}$, and the vector of realizations of benign alert type counts is $\mathbf{Z} = \{Z_1, \ldots, Z_{|\mathcal{T}|}\}$. Then we have

$$B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z}) = \max \left\{ B - \sum_{i=1}^{o(t)-1} \min \left\{ b_{o_i}, Z_{o_i} C_{o_i} \right\}, 0 \right\}. \tag{3.1}$$

Now, let us take a moment to unpack this expression for context. For the audited alert type t, we repeatedly compare the threshold b_t with $Z_t C_t$ to determine how much budget will be left for the types that follow in the priority order. If the total budget that is eaten by inspecting alerts prior to t is larger than B, $B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})$ returns 0, and no alerts of type t will be inspected. Next, we can compute the number of alerts of type t that are audited as

$$n_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z}) = \min \left\{ \lfloor B_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})/C_t \rfloor, \lfloor b_t/C_t \rfloor, Z_t \right\}. \tag{3.2}$$

Suppose that an attack generates an alert of type t. As noted earlier, we assume that the number of alerts generated due to attacks is a negligible proportion of all generated alerts (e.g., when P_e are small). Then, the probability that an alert of type t generated through an

attack is detected is approximately

$$P_{al}(\boldsymbol{o}, \mathbf{b}, t) \approx \mathbb{E}_{\mathbf{Z}}\left[\frac{n_t(\boldsymbol{o}, \mathbf{b}, \mathbf{Z})}{Z_t}\right]. \tag{3.3}$$

The approximation comes from the fact that we use the benign counts Z_t in the denominator to approximate the sum of the number of the false positive alerts and the true positive alerts in type t. This is because 1) the number of true positive alerts in each type is very small in practice and 2) the exact number true positives are unknown to the auditor.

The adversary e does not directly choose alert types, but rather the victims v (e.g., an EHR). The probability of detecting an attack $\langle e, v\rangle$ under audit order $\boldsymbol{o}$ and audit thresholds $\mathbf{b}$ is then

$$P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle) = \sum_t P^t(\langle e, v\rangle) P_{al}(\boldsymbol{o}, \mathbf{b}, t). \tag{3.4}$$

We now have sufficient preliminaries to define the utility functions of the adversaries $e \in \mathcal{E}$. Let $M(\langle e, v\rangle)$ denote the penalty of the adversary when captured by the auditor, $R(\langle e, v\rangle)$ denote the benefit if the adversary is not audited, and $K(\langle e, v\rangle)$ the cost of an attack. The utility of the adversary is then

$$\begin{aligned} U_a(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle) = {} & P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle) \cdot M(\langle e, v\rangle) \\ & + (1 - P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle)) \cdot R(\langle e, v\rangle) - K(\langle e, v\rangle). \end{aligned} \tag{3.5}$$

By assuming that the game is zero-sum, there is no difference between the Strong Stackelberg Equilibrium (SSE) and the Nash Equilibrium (NE) [119]. Under this assumption, the auditor's goal can be transferred into finding a randomized strategy $\boldsymbol{p}$ and type-specific thresholds $\mathbf{b}$ to minimize the expected utility of the adversary:

$$\min_{\boldsymbol{p}, \mathbf{b}} \quad \sum_{e \in \mathcal{E}} P_e \max_v \sum_{\boldsymbol{o} \in \boldsymbol{O}} p_{\boldsymbol{o}} U_a(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle), \tag{3.6}$$

where $\boldsymbol{p} = \{p_{\boldsymbol{o}} \mid \boldsymbol{o} \in \boldsymbol{O}\}$. We call this optimization challenge the *optimal auditing problem (OAP)*.

The optimal auditing policy can be computed using the following mathematical program, which directly extends the standard linear programming formulation for computing mixed-strategy Nash equilibria in zero-sum games:

$$\begin{aligned} &\min_{\mathbf{b},\boldsymbol{p},\mathbf{u}} \ \textstyle\sum_{e\in\mathcal{E}} P_e u_e \\ &s.t. \quad \forall \langle e, v\rangle, \ \ u_e \geq \textstyle\sum_{\boldsymbol{o}\in\boldsymbol{O}} p_{\boldsymbol{o}} U_a(\boldsymbol{o}, \mathbf{b}, \langle e, v\rangle) \\ &\qquad \textstyle\sum_{\boldsymbol{o}\in\boldsymbol{O}} p_{\boldsymbol{o}} = 1, \\ &\qquad \forall \boldsymbol{o} \in \boldsymbol{O}, \ \ 0 \leq p_{\boldsymbol{o}} \leq 1. \end{aligned} \tag{3.7}$$

An important issue in this formulation is that we do not randomize over the decision variables $\mathbf{b}$. However, if we restrict strategies to the decision variables $\boldsymbol{p}$ by fixing $\mathbf{b}$ first, then the resulting SSE and NE are identical. Indeed, if we fix $\mathbf{b}$, the formulation becomes a linear program. Nevertheless, since the set of all possible alert prioritizations is exponential, even this linear program has exponentially many variables. Furthermore, introducing decision variables $\mathbf{b}$ makes it non-linear and non-convex. Next, we show that solving this problem is NP-hard, even in a restricted special case. We prove this by reducing from the 0-1 Knapsack problem.

Definition 1 (0-1 Knapsack Problem) *Let $\mathcal{I}$ be a set of items where each item $i \in \mathcal{I}$ has a weight w_i and a value v_i, with w_i and v_i integers. W is a budget on the total amount of weight (an integer). Question: given a threshold K, does there exist a subset of items $R \subseteq \mathcal{I}$ such that $\sum_{i\in R} v_i \geq K$ and $\sum_{i\in R} w_i \leq W$?*

Theorem 1 *OAP is NP-hard even when* $\mathbf{O}$ *is a singleton.*

Proof We reduce from the 0-1 Knapsack problem defined by Definition 1. We begin by constructing a special case of the auditing problem and work with the decision version of optimization Equation 3.6, in which we decide whether the objective is below a given threshold θ. First, suppose that $Z_t = 1$ for all alert types $t \in \mathcal{T}$ with probability 1. Since the set of orders is a singleton, the probability distribution over orders $p_{\boldsymbol{o}}$ is not relevant. Consequently, it suffices to consider $b_t \in \{0,1\}$ for all t, and the actual order over types is not relevant because $Z_t = 1$ for all types. Consequently, we can choose $\mathbf{b}$ to select an arbitrary subset of types to inspect subject to the budget constraint B (i.e., type t will be audited iff $b_t = 1$). Thus, the choice of $\mathbf{b}$ is equivalent to choosing a subset of alert types $A \subseteq \mathcal{T}$ to audit.

Suppose that $\mathcal{V} = \mathcal{T}$, and each victim $v \in \mathcal{V}$ deterministically triggers some alert type $v \in \mathcal{V} = \mathcal{T}$ for any attacker e. Let $M(\langle e, v\rangle) = C(\langle e, v\rangle) = 0$ for all $e \in \mathcal{E}, v \in \mathcal{V}$, and suppose that for every e, there is a unique type $t(e)$ with $R(\langle e, v\rangle) = 1$ if and only if $v = t(e)$ and 0 otherwise. Then $\max_v U_a(\mathbf{o}, \mathbf{b}, \langle e, v\rangle) = 1$ if and only if $b_{t(e)} = 0$ (i.e., alert type $t(e)$ is not selected by the auditor) and 0 otherwise. Finally, we let $P_e = 1$ for all e.[2]

For the reduction, suppose we are given an instance of the 0-1 Knapsack problem. Let $\mathcal{T} = \mathcal{I}$, and for each $i \in \mathcal{I}$, generate v_i attackers with $t(e) = i$. Thus, $v_i = |\{e : t(e) = i\}|$. Let $C_i = w_i$ be the cost of auditing alerts of type i, and let $B = W$. Define $\theta = |\mathcal{E}| - K$. Now observe that the objective in Equation 3.6 is below θ if and only if $\min_{\mathbf{b}} \sum_{t:b_t=0} v_t \leq \theta$, or, equivalently, if there is R such that $\sum_{t \in R} v_t \geq K$. Thus, the objective of Equation 3.6 is below θ if and only if the Knapsack instance has a subset of items $R \subseteq \mathcal{I}$ which yield $\sum_{i \in R} v_i \geq K$, where R must satisfy the same budget constraint in both cases. ■

[2] While this is inconsistent with our assumption that attackers constitute only a small portion of the system users, we note that this is only a tool for the hardness proof.

3.3 Solving the Alert Prioritization Game

There are two practical challenges that need to be addressed to compute useful approximate solutions to the OAP. First, there is an exponential set of possible orderings of alert types that need to be considered to compute an optimal randomized strategy for choosing orderings. Second, there is a combinatorial space of possible choices for the threshold vectors $\mathbf{b}$. In this section, we develop a column generation approach for the linear program induced when we fix a threshold vector $\mathbf{b}$. We then introduce a search algorithm to compute the auditing thresholds.

3.3.1 Column Generation Greedy Search

By fixing the auditing threshold vector $\mathbf{b}$, Equation 4.3 becomes a linear program, albeit with an exponential number of variables. However, since the number of constraints is small, only a limited number of variables will be non-zero. In other words, the number of effective orderings of alert types in the optimal solution is small compared to the exponential search space. The challenge is in finding this small basis. We solve this problem in a greedy manner by applying the column generation framework. In this approach, we iteratively solve a linear program, where we use a subset of the variables. Upon each iteration we add a new variable before the value of the objective function fails to reduce. By doing so, we can incorporate the orderings that contribute to reducing the value of the objective function. When no new orderings can be added, the process terminates. We refer to this method as Column Generation Greedy Search (CGGS), the pseudocode for which is in Algorithm 1.

Specifically, we begin with a small subset of alert prioritizations $\boldsymbol{Q} \subseteq \boldsymbol{O}$. We solve the linear program induced after fixing $\mathbf{b}$ in Equation 4.3, restricted to columns in $\boldsymbol{Q}$. For reference purposes, we call this the *master problem*, which is generated by function $G_{lp}(*)$. Next, we check if there exists a column (ordering over types) that improves upon the current best solution. The column of parameter matrix of constraints can be denoted as $\Gamma_{p_{\boldsymbol{o}}} =$

$P_{at}(\boldsymbol{o}, \mathbf{b}, \langle e, v \rangle) - \mathbf{1}$ for the decision variable $p_{\boldsymbol{o}}$ or $\Gamma_{u_e} = \mathbf{1}$ for the decision variable u_e. The corresponding reduced costs, computed by function $rc(*)$, are $\mathbf{C^r}_{p_{\boldsymbol{o}}} = 1 - \boldsymbol{\pi_Q} \cdot \Gamma_{p_{\boldsymbol{o}}}$ and $\mathbf{C^r}_{u_e} = -\boldsymbol{\pi_Q} \cdot \Gamma_{u_e}$, where $\boldsymbol{\pi_Q}$ is the solution of the dual problem. By minimizing the reduced costs, we generate one new column in each iteration and add it to the subset of columns $\boldsymbol{Q}$ in the master problem. Within the process of generating a new column, we use $\Gamma'_{(\boldsymbol{o'}+\mathbf{t})}$ to denote the parameter column with the audit order $(\boldsymbol{o'}+t)$. This process is repeated until we can prove that the minimum reduced cost is non-negative.

The subproblem of generating the optimal column is itself non-trivial. We address this subproblem through the application of a greedy algorithm for generating a reduced-cost-minimizing ordering over alert types. The intuition behind CGGS is that, in the process of generating a new audit order, we greedily add one alert type at a time to minimize the reduced cost *given the order generated thus far*. We continue until the objective (reduced cost) fails to improve.

ALGORITHM 1: Column Generation Greedy Search (CGGS)

Input : The set $\boldsymbol{Q}$ with a single random pure strategy for the auditor.
Output: The set of pure strategies $\boldsymbol{Q}$.

while *True* **do**
  $Z = G_{lp}(\boldsymbol{Q})$; `/* Construct LP using current` $\boldsymbol{Q}$ `*/`
  $\boldsymbol{\pi_Q} = LP(Dual(Z))$; `/* Solve dual problem */`
  $\boldsymbol{o'} = []$;
  while $|\boldsymbol{o'}| < |T|$ **do**
    $\boldsymbol{o'} = \boldsymbol{o'} + \text{argmax}_{t \in T \setminus \boldsymbol{o'}} \boldsymbol{\pi_Q} \cdot \Gamma'_{(\boldsymbol{o'}+\mathbf{t})}$;
  end
  if $\min rc(\boldsymbol{Q}) < 0$ **then**
    $\boldsymbol{Q} = \boldsymbol{Q} + \boldsymbol{o'}$;
  else
    break;
  end
end

3.3.2 Iterative Shrink Heuristic Method

Armed with an approach for solving the linear program induced by a fixed budget threshold vector $\mathbf{b}$, we now develop a heuristic procedure to find alert type thresholds.

Now, let us characterize the range of each element in $\mathbf{b}$. First, it should be recognized that $\sum_t b_t \geq B$ because to allow otherwise would clearly waste auditing resources. Yet there is no explicit upper bound on the thresholds. However, given the distribution of the number of alerts Z_t for an alert type t, we can obtain an approximate upper bound on b_t, where $F_t(b_t/C_t) \approx 1$. This is possible because setting the thresholds above such bounds would lead to negligible improvement. Consequently, searching for a good solution can begin with a vector of audit thresholds, such that for each b_t, $F_t(b_t/C_t) \approx 1$. Leveraging this intuition, we design a heuristic method, which iteratively shrinks the values of a good[3] subset of audit thresholds according to a certain step size ε. We refer to this as the Iterative Shrink Heuristic Method (ISHM), the pseudocode for which is provided in Algorithm 2.

In each atomic searching action, ISHM first makes a subset of thresholds b_t strategically shrink. Next, it checks to see if this results in an improved solution. We introduce a variable l_h, which indicates the level (or the size) of the given subset of $\mathbf{b}$, and $\varepsilon \in (0,1)$, which controls the step size.

In the beginning, the vector of audit thresholds $\{\hat{H}_o\}$ is initialized with the approximate upper bounds. Then, by assigning $l_h = 1$, we consider shrinking each of the audit thresholds $\hat{H}_i$. The coefficient for shrinking is defined by the *ratio* in line 7, which is instantiated with the predefined step size ε; i.e., $i = 1$. If the best value for the objective function in the candidate subsets at $l_h = 1$ after shrinking shows an improvement, then the shrink is accepted and the shrinking coefficient is made smaller by increasing i. When no coefficient leads to improvement, we increase l_h by one, which induces tests of threshold combinations at the same shrinking ratio. This logic is described in line 6 through 20.

[3] "Good" in this context means that shrinking the thresholds within the subset improves the value of the objective function.

Once an improvement occurs, the search course resets based on the current **b**. The search terminates once $l_h > |T|$.

Note that for a single improvement, the worst-case time complexity is $O(\lceil 1/\varepsilon \rceil \cdot O(LP) \cdot 2^{|T|})$. Though exponential, our experiments show that ISHM achieves outstanding performance, both in terms of precision (of approaching the optimal solution) and efficiency.

ALGORITHM 2: Iterative Shrink Heuristic Method (ISHM)

Input : Instance of the game, step size ε.
Output: Vector of audit thresholds $\{\hat{H}_i\}$.

Initialize $\{\hat{H}_i\}$ with full coverages in $\{F_t\}$;
$l_h = 1$; $obj = +\infty$;
while $l_h <= |T|$ **do**
    $C_{l_h} = choose(|T|, l_h)$; `/* Find combinations */`
    $prgrs = 0$;
    for $i \leftarrow 1$ **to** $\lceil 1/\varepsilon \rceil$ **do**
        $ratio = \max\{0, 1 - i * \varepsilon\}$;
        $obj_r = +\infty$; $pst_r = 0$;
        for $j \leftarrow 1$ **to** $|C_{l_h}|$ **do**
            $temp = \{\hat{H}_i\}$;
            for $k \leftarrow 1$ **to** l_h **do**
                $temp(1, C_{l_h(j,k)}) = temp(1, C_{l_h(j,k)}) * ratio$;
            end
            $obj' = LP(B, temp)$; `/* Return LP objective value */`
            if $obj' < obj_r$ **then** $obj_r = obj'$; $pst_r = j$;
        end
        if $obj_r < obj$ **then**
            $obj = obj_r$;
            $S_u = C_{l_h}(pst_r, :)$; `/* Types in need of update */`
            for $j \leftarrow 1$ **to** $|S_u|$ **do** $\hat{H}_{S_{u_j}} = \hat{H}_{S_{u_j}} * ratio$;
            break;
        end
        $prgrs = i$;
    end
    if $prgrs == \lceil 1/\varepsilon \rceil$ **then** $l_h = l_h + 1$;
    else $l_h = 1$;
end

3.4 Controlled Evaluation

To gain intuition into the potential for our methods, we evaluated the performance of the ISHM and CGGS approaches using a synthetic dataset, *Syn_A*. To enable comparison with an optimal solution, we use a relatively small synthetic dataset, but as will be clear, it is sufficient to illustrate the relationship between our methods and the optimal brute force solution.

To perform the analysis, we vary the audit budgets B and step size ε of ISHM. In addition, we evaluate a combination of CGGS+ISHM (since the former is also an approximation), by again comparing to the optimal.

3.4.1 Data Overview

The dataset *Syn_A* consists of 5 potential attackers who perform accesses ($p_e = 1$[4]), 8 files, 4 predefined alert types, and a set of rules for triggering alerts if any access happens. Table 3.2 summarizes the information of *Syn_A* and related parameters in the corresponding scenario. We let the number of alerts for all types be distributed according to a Gaussian distribution with means and standard deviation as reported in Table 3.2a. Since the number of alerts for each alert type are integers, we discretize the x-axis of each alerts cumulative distribution function and use the corresponding probabilities for each possible alert count. We consider the 99.5 probability coverage for each alert type to obtain a finite upper bound on alert counts.

We assume alerts are triggered deterministically for each access, a common case in rule-based systems. The alert type that will be triggered for each potential access is provided in Table 3.2b, where "-" represents a benign access. This table is generated with a probability vector $[0.07, 0.38, 0.23, 0.16, 0.16]$ for each employee, which corresponds to alert type vector $[0, 1, 2, 3, 4]$. Although in reality, benign accesses may be more frequent,

[4]The artificially high incidence of attacks here is merely to facilitate a comparison with a brute-force approach.

we lower their probability to better differentiate the final value of the objective function. The benefit of the adversary for a successful attack, the cost of an attack and the cost of an audit are all directly related to the alert type, which is shown in Table 3.2a. In addition, the penalty for being caught is set to a constant value of 4.

Table 3.2: Description of Dataset *Syn_A*.

(a) Parameters for alert types in the synthetic setting.

	Type 1	**Type 2**	**Type 3**	**Type 4**
Mean	6	5	4	4
Std	2	1.6	1.3	1
99.5% Coverage	+/-5	+/-4	+/-3	+/-3
Benefit	3.4	3.7	4	4.3
Attack Cost	0.4	0.4	0.4	0.4
Audit Cost	1	1	1	1

(b) Rules for alert types in the synthetic setting.

Employee	**Record**							
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
e_1	–	3	2	2	3	4	3	1
e_2	1	–	1	1	1	2	1	1
e_3	1	3	4	–	1	3	1	4
e_4	2	1	3	1	4	4	2	2
e_5	2	3	1	4	2	1	3	2

3.4.2 Optimal Solution with Varying Budget

Based on the given information, we can compute the optimal OAP solution. First, the search space for audit thresholds in this scenario is as follows: 1) for each alert type, the audit threshold $b_t \in \mathbf{N}$, 2) the sum of thresholds for all alert types should be greater than or equal to B, 3) for each type, the upper bound of the audit threshold b_t is where $F_t(b_t/C_t) \approx 1$. Concretely, we set vector $J = Mean + |99.5\%Coverage|$ as the upper bound for finding the optimal solution. Thus, the space of the investigation of the optimal solution is $O(\prod_{i=1}^{|T|}(J_i+1))$. Note that 0 is also a possible choice, which means the auditor will not

check the corresponding alert type. Thus, it is infeasible to directly solve the OAP in the instances with a large number of alert types or large J_i.

To investigate the performance of the proposed audit model, we allocated a vector of audit budgets $\mathbf{B} = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, which has a wide range with respect to the scale of the means of the alert types. We then apply a brute force search to discover an optimal vector of budget thresholds for each type. Table 3.3 shows the optimal solution of OAP for each candidate $\mathbf{B}$, including the optimal value of the objective function, optimal threshold (using the smallest optimal threshold whenever the optimal solution is not unique), pure strategies in the support of the optimal mixed strategy, and the optimal mixed strategy of the auditor. As expected, it can be seen that as the budget increases, the optimal value of the objective function (minimized by the auditor) decreases monotonically.

Table 3.3: The optimal solution for the auditor under various budgets.

ID	Budget	Optimal Objective Value	Optimal Threshold	Effective Pure Strategy	Optimal Mixed Strategy
1	2	12.2945	[1,1,1,1]	[2,3,4,1][4,1,3,2][4,2,3,1][4,3,2,1]	[0.3566, 0.3780, 0.1210, 0.1444]
2	4	7.7176	[2,1,1,2]	[1,2,3,4][2,1,3,4][4,2,1,3][4,2,3,1]	[0.4664, 0.0052, 0.0934, 0.4350]
3	6	3.2651	[2,2,2,2]	[2,1,3,4][4,1,3,2][4,2,1,3][4,2,3,1]	[0.2748, 0.2341, 0.3293, 0.1618]
4	8	-0.4517	[3,3,2,2]	[2,1,3,4][4,1,3,2][4,2,1,3][4,2,3,1]	[0.0762, 0.4600, 0.1329, 0.3309]
5	10	-2.1314	[3,3,3,3]	[1,2,3,4][1,4,3,2][4,1,2,3][4,1,3,2]	[0.3926, 0.0788, 0.4080, 0.1206]
6	12	-3.7345	[4,4,3,3]	[2,1,3,4][4,2,3,1][4,2,1,3][4,1,3,2]	[0.2028, 0.1554, 0.2076, 0.4342]
7	14	-5.1645	[5,4,3,3]	[2,1,3,4][4,2,3,1][4,2,1,3][4,1,3,2]	[0.3559, 0.2199, 0.3176, 0.1066]
8	16	-6.4510	[6,5,4,4]	[2,1,3,4][4,1,3,2][4,2,1,3][4,2,3,1]	[0.2431, 0.2636, 0.1728, 0.3205]
9	18	-7.4649	[7,6,5,5]	[2,1,3,4][4,1,3,2][4,2,1,3][4,2,3,1]	[0.2710, 0.2630, 0.2054, 0.2615]
10	20	-8.1561	[9,7,6,6]	[1,2,3,4][4,1,2,3][4,1,3,2][4,2,3,1]	[0.2398, 0.1742, 0.2275, 0.3585]

3.4.3 Findings

Our heuristic methods aim to find an approximate solution through major reductions in computation complexity. In this respect, the search step size ε is a key factor to consider because it could lead the search into a locally optimal solution. To investigate the gap between the objective function with the optimal solution, as well as the influence of ε on the gap, we performed experiments with a series of step sizes $\varepsilon = [0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$. Tables 3.4 and 3.5, summarize the results, where each cell consists of two items: 1) the minimized sum of the maximal utilities of all adversaries obtained using the heuristic method and 2) the corresponding audit threshold vector.

There are three findings worth highlighting. First, when ε is fixed, the approximated values of the objective function decrease as the budget increases. This is akin to the trend shown in Table 3.3. Second, when the budget B is fixed, the approximated values achieved through ISHM and ISHM+CGGS exhibit a general growth trend as ε increases. This occurs because larger shrink ratios increase the likelihood that the heuristic search will miss more of the good approximate solutions. Third, we find that the ISHM and ISHM+CGGS solutions are close to the optimal. To measure the solution quality as a function of ε, we use $\gamma_\varepsilon = \frac{1}{|\mathbf{B}|}\sum_i^{|\mathbf{B}|} |\hat{S}_{\mathbf{B}_i,\varepsilon} - S_{\mathbf{B}_i,\varepsilon}|/|S_{\mathbf{B}_i,\varepsilon}|$, where $\hat{S}_{\mathbf{B}_i,\varepsilon}$ denotes the approximate optimal values in Tables 3.4 and 3.5 and $S_{\mathbf{B}_i,\varepsilon}$ denotes the optimal values provided by Table 3.3.

In Table 3.6, it can be seen that ISHM (and solving the linear program to optimality) achieves solutions near 99% of the optimal (as denoted by γ_ε^1) when the step size $\varepsilon \leq 0.2$. Even the approximately optimal solutions with $\varepsilon = 0.5$ have a good approximation ratio (above 89%). As such, it appears that if we choose an appropriate ε, then ISHM can perform well.

When we combine ISHM+CGGS (denoted by γ_ε^2), the approximation quality drops compared to γ_ε^1, as we would expect, with the lone exception of ($\varepsilon = 0.4$). However, γ_ε^2 is very close to γ_ε^1, which suggests that our approximate column generation method does not significantly degrade the quality of the solution.

Next, we consider the computational burden for ISHM to achieve an approximate target of the optimal solution. Table 3.7 provides the values of the threshold vectors under various B and ε. It can be seen that the number of threshold candidates explored decreases as the step size grows. For a given ε, the number of thresholds considered by the algorithm initially increases, but then drops as the audit budget increases. The reason that less effort is necessary at the extremes of the budget range is that the restart of the test for a single alert type (to find a better position) is invoked less frequently. By contrast, a larger amount of effort is required in the middle of the budget range due to more frequent restarts (although this yields only a small improvement).

Finally, we investigate the average number for the threshold vectors explored by the algorithm over the budget range $\mathbf{B}$. For the various step sizes, we represent the results in vector form $\mathscr{T} = [403, 223, 156, 121, 93, 86, 68, 66, 61, 47]$. Dividing by the number of investigations needed to discover the optimal solution, the resulting ratio vector is $\mathscr{T}' = [0.0831,\ 0.0460,\ 0.0321,\ 0.0251,\ 0.0198,\ 0.0190,\ 0.0163,\ 0.0182,\ 0.0206,\ 0.0210]$. Thus, when $\varepsilon = 0.2$ (when both γ_ε^1 and γ_ε^2 are greater than 0.99), the number of thresholds explored is only 2.51% of the entire space. As such, by applying ISHM, the number of investigated threshold candidates can be greatly reduced without significantly sacrificing solution quality.

Table 3.4: The approximation of the optimal solutions obtained by ISHM at various levels of B and ε.

B	**Approximation of Optimal Loss of the Auditor and corresponding thresholds by ISHM**									
	$\varepsilon = 0.05$	$\varepsilon = 0.10$	$\varepsilon = 0.15$	$\varepsilon = 0.20$	$\varepsilon = 0.25$	$\varepsilon = 0.30$	$\varepsilon = 0.35$	$\varepsilon = 0.40$	$\varepsilon = 0.45$	$\varepsilon = 0.50$
2	12.2945 [10,1,1,1]	12.2945 [9,1,1,1]	12.2958 [9,9,1,1]	12.2945 [8,1,1,1]	12.2958 [8,9,1,1]	12.3675 [7,9,7,7]	12.3675 [7,9,7,7]	12.2945 [6,1,1,1]	12.3675 [6,9,7,7]	12.3675 [5,9,7,7]
4	7.7176 [2,1,1,2]	7.7176 [2,1,1,2]	7.7176 [2,1,1,2]	7.7176 [2,1,1,2]	7.7176 [2,1,1,2]	7.7176 [2,1,1,2]	7.7181 [2,1,7,2]	7.8402 [1,1,7,7]	7.8402 [1,9,1,3]	7.9037 [11,9,1,3]
6	3.2651 [2,2,2,2]	3.2651 [2,2,2,2]	3.2651 [2,2,2,2]	3.2651 [2,2,2,2]	3.2651 [2,2,2,2]	3.2651 [2,2,2,2]	3.3267 [3,3,2,2]	3.2744 [2,3,2,2]	3.4549 [11,2,3,3]	3.4549 [11,2,3,3]
8	−0.4517 [3,3,2,2]	−0.4517 [3,3,2,2]	−0.4517 [3,3,2,2]	−0.4517 [3,3,2,2]	−0.4517 [3,3,2,2]	−0.3508 [4,4,2,2]	−0.4517 [3,3,2,2]	−0.4116 [11,3,2,2]	−0.3730 [3,4,3,3]	−0.2910 [5,4,3,3]
10	−2.1314 [3,3,3,3]	−2.1314 [3,3,3,3]	−2.1314 [3,3,3,3]	−2.1314 [3,3,3,3]	−2.1314 [3,3,3,3]	−1.9693 [4,4,4,4]	−1.9996 [4,3,4,4]	−2.0119 [3,3,4,4]	−2.0755 [3,4,3,3]	−2.0037 [5,4,3,3]
12	−3.7345 [4,4,3,3]	−3.7345 [4,4,3,3]	−3.7345 [4,4,3,3]	−3.7345 [4,4,3,3]	−3.7345 [4,4,3,3]	−3.5991 [4,4,4,4]	−3.5627 [4,5,4,4]	−3.4854 [6,5,4,4]	−3.6533 [6,4,3,3]	−3.6873 [5,4,3,3]
14	−5.0713 [9,4,3,5]	−5.0713 [9,4,3,5]	−5.0430 [11,5,3,3]	−5.0430 [11,5,3,3]	−5.0713 [5,4,3,5]	−5.0962 [7,4,4,4]	−5.0350 [7,5,4,4]	−5.0629 [6,5,4,4]	−5.0713 [6,4,3,7]	−5.0713 [5,4,3,7]
16	−6.4510 [6,5,4,4]	−6.4510 [6,5,4,4]	−6.4363 [7,5,4,4]	−6.4510 [6,5,4,4]	−6.3823 [6,6,5,5]	−6.4135 [7,6,4,4]	−6.4363 [7,5,4,4]	−6.4510 [6,5,4,4]	−6.3225 [6,9,7,7]	−6.1149 [5,9,7,7]
18	−7.4649 [7,6,5,5]	−7.4649 [7,6,5,5]	−7.4600 [7,7,5,5]	−7.4490 [8,7,5,5]	−7.4585 [8,6,5,5]	−7.4490 [7,6,7,7]	−7.4320 [7,9,7,7]	−7.3956 [11,9,7,4]	−7.3612 [6,9,7,7]	−6.1149 [5,9,7,7]
20	−8.1561 [9,7,6,6]	−8.1561 [9,7,6,6]	−8.1548 [9,7,7,7]	−8.1523 [8,7,7,7]	−8.1520 [8,9,7,7]	−8.1308 [11,6,7,7]	−8.1138 [7,9,7,7]	−7.6619 [11,9,7,4]	−7.3612 [6,9,7,7]	−6.1149 [5,9,7,7]

Table 3.5: The approximation of the optimal solutions obtained by ISHM + CGGS at various levels of B and ε.

B	**Approximation of Optimal Loss of the Auditor and corresponding thresholds by ISHM + CGGS**									
	$\varepsilon = 0.05$	$\varepsilon = 0.10$	$\varepsilon = 0.15$	$\varepsilon = 0.20$	$\varepsilon = 0.25$	$\varepsilon = 0.30$	$\varepsilon = 0.35$	$\varepsilon = 0.40$	$\varepsilon = 0.45$	$\varepsilon = 0.50$
2	12.2967 [1,1,1,1]	12.2967 [1,1,1,1]	12.3096 [9,9,1,1]	12.2967 [1,1,1,1]	12.3096 [8,9,1,1]	12.3677 [7,9,7,7]	12.3677 [7,9,7,7]	12.2967 [1,1,1,1]	12.3677 [6,9,7,7]	12.3677 [5,9,7,7]
4	7.7214 [2,1,1,2]	7.7214 [2,1,1,2]	7.7346 [2,9,1,2]	7.7214 [2,1,1,2]	7.7346 [2,9,1,2]	7.7346 [2,9,1,2]	7.7346 [2,9,1,2]	7.9151 [1,1,1,7]	7.8402 [1,9,1,3]	7.9045 [11,9,1,3]
6	3.2755 [2,2,2,2]	3.2755 [2,2,2,2]	3.2755 [2,2,2,2]	3.2755 [2,2,2,2]	3.2755 [2,2,2,2]	3.2755 [2,2,2,2]	3.3628 [3,3,2,2]	3.3267 [2,3,2,2]	3.4897 [11,2,3,3]	3.3099 [2,2,3,3]
8	−0.4422 [3,3,2,2]	−0.4422 [3,3,2,2]	−0.4422 [3,3,2,2]	−0.4422 [3,3,2,2]	−0.2761 [5,2,2,7]	−0.3300 [4,4,2,2]	−0.4006 [4,3,2,2]	−0.4422 [3,3,2,2]	−0.3404 [3,4,3,3]	−0.2761 [5,2,3,3]
10	−2.1203 [3,3,3,3]	−2.1203 [3,3,3,3]	−2.1203 [3,3,3,3]	−2.1203 [3,3,3,3]	−2.1203 [3,3,3,3]	−1.9503 [4,4,4,4]	−1.9873 [4,3,4,4]	−2.0091 [3,3,4,4]	−2.0612 [3,4,3,3]	−1.9508 [5,4,3,3]
12	−3.7215 [4,4,3,3]	−3.7215 [4,4,3,3]	−3.7215 [4,4,3,3]	−3.7215 [4,4,3,3]	−3.7215 [4,4,3,3]	−3.5832 [4,4,4,4]	−3.5448 [4,5,4,4]	−3.4326 [6,5,4,4]	−3.6383 [6,4,3,3]	−3.6768 [5,4,3,3]
14	−5.0709 [5,9,3,4]	−5.1529 [5,4,4,4]	−5.0430 [9,4,3,3]	−5.0700 [6,5,3,4]	−5.0698 [6,4,3,7]	−5.0857 [7,4,4,4]	−5.0125 [7,5,4,4]	−5.0494 [6,5,4,4]	−5.0698 [6,4,3,7]	−5.0706 [5,4,3,7]
16	−6.4394 [6,5,4,4]	−6.4394 [6,5,4,4]	−6.4258 [7,5,4,4]	−6.4394 [6,5,4,4]	−6.3683 [6,6,5,5]	−6.4008 [7,6,4,4]	−6.4258 [7,5,4,4]	−6.4394 [6,5,4,4]	−6.3038 [6,9,7,7]	−6.1149 [5,9,7,7]
18	−7.4524 [7,6,5,5]	−7.4524 [7,6,5,5]	−7.4465 [7,7,5,5]	−7.4363 [8,7,5,5]	−7.4472 [8,6,5,5]	−7.4359 [7,6,7,7]	−7.4171 [7,9,7,7]	−7.3825 [11,5,7,7]	−7.3612 [6,9,7,7]	−6.1149 [5,9,7,7]
20	−8.1448 [9,7,6,6]	−8.1448 [9,7,6,6]	−8.1433 [9,7,7,7]	−8.1398 [8,7,7,7]	−8.1388 [8,9,7,7]	−8.1207 [11,6,7,7]	−8.1043 [7,9,7,7]	−7.6619 [11,9,7,4]	−7.3612 [6,9,7,7]	−6.1149 [5,9,7,7]

Table 3.6: The average precision over the budget vector **B** by applying ISHM and ISHM+CGGS.

ε	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
γ_ε^1	0.9982	0.9982	0.9973	0.9974	0.9970	0.9634	0.9830	0.9680	0.9549	0.8982
γ_ε^2	0.9943	0.9959	0.9932	0.9940	0.9560	0.9562	0.9684	0.9700	0.9452	0.8966

Table 3.7: The number of threshold vectors checked by ISHM with a given budget B and step size ε.

ε	B									
	2	4	6	8	10	12	14	16	18	20
0.10	251	267	255	243	235	227	199	207	191	171
0.20	128	144	148	140	132	124	108	108	92	84
0.30	65	109	101	93	85	85	81	77	69	65
0.40	74	66	78	70	70	62	62	62	50	50
0.50	35	43	47	47	47	47	43	35	35	35

3.5 Model Evaluation

The previous results suggest ISHM and CGGS can be efficient and effective in solving the OAP in a small controlled environment. Here, we investigate the performance of the proposed game-theoretical audit model on more realistic and larger datasets. This evaluation consists of comparing the quality of solutions of OAP with several natural alternative auditing strategies.

The first dataset, *Rea_A*, corresponds to the EHR access logs of Vanderbilt University Medical Center (VUMC). This dataset is notable because VUMC privacy officers rely on this data to conduct retrospective audits to determine if there are accesses that violate organizational policy. The central goal in this use case is to preserve patient privacy. Given that this is not a publicly available dataset, we conducted experiments with a second dataset, *Rea_B*, which consists of public observations of credit card applications. It labels applicants as having either low or high risk of fraud. We provide an audit mechanism to capture events of credit card fraud based on the features in this dataset. We use *Rea_B* to demonstrate the broad applicability of the proposed approaches and enable replication of our results.

Table 3.8: Description of the EHR alert types.

ID	Alert Type Description	Mean	Std
1	Same Last Name	183.21	46.40
2	Department Co-worker	32.18	23.14
3	Neighbor ($\leq$ 0.5 miles)	113.89	80.44
4	Last Name; Same address	15.43	14.61
5	Last Name; Neighbor ($\leq$ 0.5 miles)	23.75	11.07
6	Same address; Neighbor ($\leq$ 0.5 miles)	20.07	11.49
7	Last Name; Same address; Neighbor ($\leq$ 0.5 miles)	32.07	16.54

3.5.1 Data Overview

Rea_A consists of the VUMC EHR access logs for 28 continuous workdays during 2017. There are 48.6M access events, 38.7M (79.5%) of which are repeated accesses.[5] We filtered out the repeated accesses to focus on the distinct user-patient relationships established on a daily basis. The mean and standard deviation of daily access events was 355,602.18 and 195,144.99, respectively. The features for each event include: 1) timestamp, 2) patient ID, 3) employee ID, 4) patient's residential address, 5) employee's residential address, 6) employee's VUMC department affiliation and 7) indication of if a patient is an employee. We focus on the following alert types: 1) employee and patient share the same last name, 2) employee and patient work in the same VUMC department, 3) employee and patient share the same residential address, and 4) employee and patient are neighbors within a distance threshold.

In certain cases, the same access may generate multiple alerts, each with a distinct type. For example, if a husband, who is a BMRC employee, accesses his wife's EHR, then two alert types may be triggered: 1 (same last name) and 3 (same address). We, therefore, redefine the set of alert types to also consider combinations of alert categories. The resulting set of alert types is detailed in Table 3.8.

We label each access event in the logs with a corresponding alert type or as "benign"

[5]We define a repeated access as an access that is committed by the same employee to the same patient's EHR on the same day.

(i.e., no alerts generated). To evaluate our methods, we choose a random sample of 50 employees and patients who generate at least one alert. This set of employees and the set of patients then results in 2500 *potential* accesses, where each employee can access each patient.

We let the probability that an employee could be malicious be 1, which is artificially high, but enables us to clearly compare the methods in the experiments. The benefit vector for the adversary is $[10, 12, 12, 24, 25, 25, 27]$ for the corresponding categories of alert types (1-7 in Table 3.8). The penalty for capture is set to 15. We set the cost of both an attack and an audit to 1. We acknowledge that the model parameters are *ad hoc*, but this does not affect the results of our comparative analysis. In practice, this would be accomplished based on expert opinion, but is outside the scope of this study.

Rea_B is the Statlog (German Credit Data) dataset available from the UCI Machine Learning Repository. *Rea_B* contains 1000 credit card applications. It is composed of 20 attributes describing the status of the applicants pertaining to their credit risk. Before issuing a credit card, banks would determine if it could be fraudulent based on the features in the data. Nevertheless, no screening process is perfect, and given a large number of applications, applications will require retrospective audits to determine whether specific applications should be canceled. Thus, alerts in this setting aim to indicate potential fraud and a subset of such alerts are chosen for a time-consuming auditing process. Leveraging the provided features, we define 5 alert types, which are triggered by the specific combinations of attribute values and the purpose of the application. The 8 selected purposes of application are the "victims" in our audit model. Table 3.9 summarizes how alerts are triggered. In the description field, italicized words represent the purpose of the application, while the other words represent feature values.

We used the 5 alert categorizations discussed above to label the 1000 applications with alert types, excluding any that fail to receive a label. Among these, we randomly selected 100 applicants who may choose to "attack" one of the 8 purposes of credit card

Table 3.9: Description of the defined alert types.

ID	Alert type Description	Mean	Std
1	No checking account, *Any purpose*	370.04	15.81
2	Checking < 0, *New car, Education*	82.42	7.87
3	Checking > 0, Unskilled, *Education*	5.13	2.08
4	Checking > 0, Unskilled, *Appliance*	28.21	5.25
5	Checking > 0, Critical account, *Business*	8.31	2.96

applications, for a total of 800 possible events. The benefit vector for the adversary is $[15, 15, 14, 20, 18]$ for each of the alert types generated, respectively. We set the penalty for detection to 20 and costs for attack and audit were both set to 1. Again, to facilitate comparison we set $p_e = 1$ in all cases.

3.5.2 Comparison with Baseline Alternatives

The performance of the proposed audit model was investigated by comparing with several natural alternative audit strategies as baselines. The first alternative is to randomize the audit order over alert types, which we call *Audit with random orders of alert types*. Though random, this strategy mimics the reality of random reporting (e.g., where a random patient calls a privacy official to look into alleged suspicious behavior with respect to the use of their EHR). In this case, we adopt the thresholds out of the proposed model with $\varepsilon = 0.1$ to investigate the performance. The second alternative is to randomize the audit thresholds. We refer to this policy as *Audit with random thresholds*. For this policy, we assume that 1) the auditor's choice satisfies $\sum_i b_i \geq B$ and 2) the auditor has the ability to find the optimal audit order after deciding upon the thresholds. The third alternative is a naive greedy audit strategy, where the auditor prioritizes alert types according to their utility loss (i.e., greater consequence of violations). In this case, the auditor investigates as many alerts of a certain type as possible before moving on to the next type in the order. For our experiments, when the alert type order is based on the loss of the auditor, which is the benefit the adversary

receives when they execute a successful attack. Thus, we refer to this strategy as *Audit based on benefit*.

The following performance comparisons are assessed over a broad range of auditing budgets. For our model, we present the values of the objective function with three different instances of the step size ε in ISHM: $[0.1, 0.2, 0.3]$. Figures 4.1 and 3.2 summarize the performance of the proposed audit model and three alternative audit strategies for *Rea_A* and *Rea_B*, respectively.

For dataset *Rea_A*, the range of B was set to 10 through 100. The budget of 100 covers about $1/4$ of the sum of the means of the seven alert types. In reality, such coverage is quite high. By applying the proposed audit model, we approximately solve the OAP given B and ε. For *Audit with random orders of alert types*, we assign the audit thresholds using ISHM with $\varepsilon = 0.1$. The randomization is repeated 2000 times without replacement. As for *Audit with random thresholds*, we randomly generate the audit thresholds to solve the corresponding LP, which are repeated 5000 times. For *Audit based on benefit*, we randomly sample 2000 instances of $\mathbf{Z}$ based on the distributions of alert types learned from the dataset.

Based on Figure 4.1, there are several findings we wish to highlight. First, in our model, as the audit budget increases, the auditor's loss decreases. At the high end, when $B \geq 90$, the auditor's loss is zero, which, in the VUMC audit setting, implies that all the potential adversaries are deterred from an attack. This valuation of B is smaller than $1/4$ of the sum of distribution means of all alert types. The reason for this phenomenon stems from the fact that when the audit budget increases, the audit model finding better approximations of the optimal audit thresholds, which, in turn, enables the auditor to significantly limit the potential gains of the adversaries. Second, our proposed model significantly outperforms all of the baselines. Third, even though *Audit with random orders of alert types* uses approximated audit thresholds, the auditor's loss is substantially greater than our proposed approach. However, the auditor's losses for the alternatives approach ours when $B = 20$.

This is because the thresholds are $[0,0,0,7,0,11,8]$, such that the audit order is less of a driver than in other situations. Fourth, *Audit based on benefit* tends to have very poor performance compared to other policies. This is because when the audit order is fixed (or is predictable), adversaries have greater evasion ability and attack more effectively. Fifth, *Audit with random thresholds* tends to outperform the other baselines but is still significantly worse than our approach. The is because the auditor has the ability to search for the optimal audit policy, but the thresholds are randomly assigned such that they are hampered in achieving the best solution.

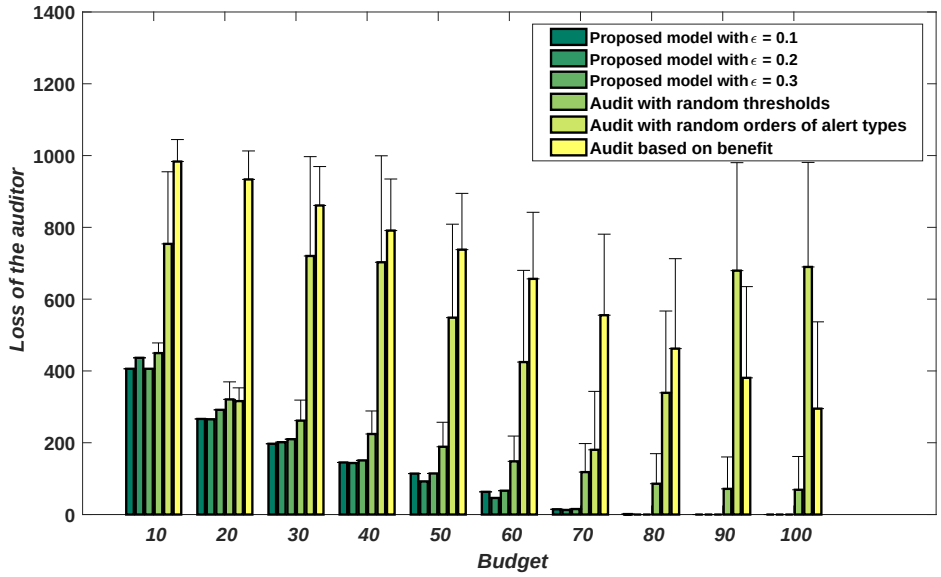


Figure 3.1: Auditor's loss in the proposed and baseline models in the *Rea_A* dataset.

For the credit card application scenario, Figure 3.2 compares the auditor's loss in our heuristics and the three baselines. For dataset *Rea_B*, the range for B is 10 to 250 with a step size of 20. As expected, as the budget increases, the auditor sustains a decreasing average loss. It can be seen that the proposed audit model significantly outperforms the alternative baselines. Specifically, as the auditing budget increases, the auditor's loss trends towards, and becomes, 0 in our approach. This means that the attackers are completely deterred. For the alternatives, as before, *Audit with random thresholds* outperforms other strategies. And, just as before, the strategy that greedily audits alert types (in order of loss) tends to

perform quite poorly.

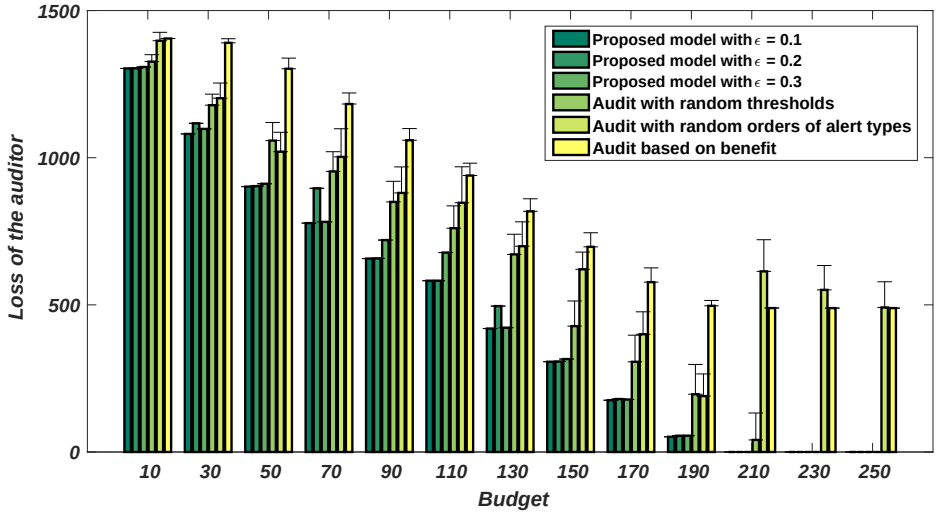


Figure 3.2: Loss of the auditor in the proposed and alternatives audit model in the *Rea_B* dataset.

3.6 Disccusion

TDMTs are usually deployed in database systems to address a variety of attacks that originate from within and beyond an organization. However, an overwhelming alert volume is far beyond the capability of auditors with limited resources. Our research illustrates that policy compliance auditing, as a significant component of database management, can be improved by prioritizing which alerts to focus on via a game theoretic framework, allowing auditing policies to make the best use of limited auditing resources while simultaneously accounting for the strategic behavior of potential policy violators. This is notable because auditing is critical to a wide range of management requirements, including privacy breach and financial fraud investigations. As such, this model and the effective heuristics we offer in this study fill a major gap in the field.

There are several limitations of our approach that we wish to highlight as opportunities for future investigations. First, there are limitations to the parameterization of the game. One notable aspect is that we assumed that the game has a zero-sum property. Yet in

reality, this may not be the case. For example, an auditor is likely to be concerned less about the cost incurred by an adversary for executing an attack and more concerned about the losses that arise from successful violations Additionally, while our experiments show the proposed audit model outperforms natural alternatives, it is unclear how sensitive this result is to parameter variations. Thus, a fruitful direction of research is in the payoff structures and how they influence the performance of the model.

A second set of limitations stems from the assumptions we rely upon. In particular, we assumed that each attack is instantaneous, which turned the problem into a one-shot two-stage game. However, attacks in the wild may require multiple cycles to fully execute, such that the auditor may be able to capture the attacker before they complete their exploit. To address such a setting, a temporal audit model may complement the approach introduced in this chapter. Furthermore, our model is predicated on an environment in which the auditor has complete knowledge, including the identities, about the set of potential adversaries. However, in practice, one player can hardly know everything about the other. Thus, a natural follow-up investigation is to relax such a strong assumption by involving uncertainty in the knowledge of the players.

A third limitation is in the economic premise of the attack. Specifically, we expected the interaction between the auditor and adversaries as fully rational. In reality, adversaries may be bounded in their rationality, and an important extension would be to generalize the model consider such behavior.

3.7 Conclusion

Prioritizing the alerts raised by TDMT modules can enable effective auditing of privacy- and security-related incidents. This chapter introduced a game theoretic model to represent the strategic interactions between an auditor and a set of potential adversaries. We showed that discovering the optimal prioritization of alerts is NP-hard, but that several efficient search heuristics can be designed to solve the problem. Using a controlled, synthetic,

dataset, we proved that the heuristics can achieve a performance that is close to the optimal solution. And, using several different types of datasets illustrated that the heuristics are substantially more effective at prioritization than typical auditing strategies invoked in practice. We did, however, make several simplifying assumptions regarding the behavior of the adversaries and the parameterization of the variables in the model, but believe that this research provides a foundation for further investigation in alert prioritization games.

Part III

Online Auditing: Signaling

Chapter 4

Signaling Audit Game: an online solution

Routine operational use of sensitive data is often governed by law and regulation. For instance, in the medical domain, there are various statues at the state and federal level that dictate who is permitted to work with patients' records and under what conditions. To screen for potential privacy breaches, logging systems are usually deployed to trigger alerts whenever a suspicious access is detected. However, such mechanisms are often inefficient because 1) the vast majority of triggered alerts are false positives, 2) small budgets make it unlikely that a real attack will be detected, and 3) attackers can behave strategically, such that traditional auditing mechanisms cannot easily catch them. To improve efficiency, information systems may invoke signaling, so that whenever a suspicious access request occurs, the system can, in *real time*, warn the user that the access may be audited. Then, at the close of a finite period, a selected subset of suspicious accesses are audited. This gives rise to an online problem in which one needs to determine 1) whether a warning should be triggered and 2) the likelihood that the data request event will be audited. In this chapter, we formalize this auditing problem as a Signaling Audit Game (SAG), in which we model the interactions between an auditor and an attacker in the context of signaling and the usability cost is represented as a factor of the auditor's payoff. We study the properties of its Stackelberg equilibria and develop a scalable approach to compute its solution. We show that a strategic presentation of warnings adds value in that SAGs realize significantly higher utility for the auditor than systems without signaling. We perform a series of experiments with 10 million real access events, containing over 26K alerts, from a large academic medical center to illustrate the value of the proposed auditing model and the consistency of its advantages over existing baseline methods.

4.1 Introduction

Our society now collects, stores, and processes personal and intimate data with ever-finer detail, documenting our activities and innovations in a wide range of domains, ranging from health to finance [120, 121]. Due to the potential value of such data, their management systems face non-trivial challenges to personal privacy and organizational secrecy. The sensitive nature of the data stored in such systems attracts malicious attackers who can gain value by disrupting them in various ways (e.g., stealing sensitive information, commandeering computational resources, committing financial fraud, and simply shutting the system down) [122, 58]. Reports in the popular media indicate that the severity and frequency of attack events continues to grow. Notably, the recent breach at Equifax led to the exposure of data on 143 million Americans, including credit card numbers, Social Security numbers, and other information that could be used for identity theft or other illicit purposes [11]. Even more of a concern is that the exploit of the system continued for at least two months before it was discovered.

To defend against attack, modern database systems are often armed with an alerting capability to detect and notify about potential risks incurred during daily use [123, 124, 125]. This entails the logging of access events, which can be thought of as a collection of rules, each of which defines a semantic type of a potentially malicious situation [126, 25]. In mission-critical systems, the access requests of authenticated users are often granted to ensure continuity of workflow and operations, such that notification about potential misuse is provided to administrators who perform retrospective audit investigations [114, 97, 127, 128]. For instance, many healthcare organizations (HCOs) rely on alert, as well auditing, mechanisms to monitor anomalous accesses to electronic health records (EHRs) by employees who may violate policy and breach the privacy of certain patients [129]. Similarly, the providers of online services, such as financial institutions and social media platforms, often use alerts and audits to defend against attacks, such as financial fraud and compromises to computational resources [130]. Though audits do not directly prevent attacks in

their own right, they allow for the discovery of breaches that can be followed up on before they escalate to full blown exploits by attackers.

However, there are challenges to instituting robust auditing schemes in practice. First, the volume of triggered alerts is typically far greater than the auditing capacity of an organization [131]. Second, in practice, the majority of triggered alerts correspond to false positives, which stem from an organization's inability to define and recognize complex dynamic workflows. Third, to mitigate the risk of being caught, attackers prefer to act strategically, such as carefully choosing the way (or target) to attack. And last, but not least, in the retrospective audit setting, attacks are not discovered until they are investigated.

In essence, this is a resource allocation problem in an adversarial environment for which the Stackelberg security game (SSG) is a natural choice to apply for modeling purposes [132, 34, 86, 133, 134, 135]. In this model, the *defender* first commits to a budget allocation policy and, subsequently, the *attacker* responds with the optimal attack based on the defender's strategy. This model has enabled the design and deployment of solutions to various security problems in practice, such as *ARMOR* (which was adopted by the Los Angeles Police Department (LAPD) to randomize checkpoints on the roadways at Los Angeles International Airport) [92] and *IRIS* (which was adopted by the US Federal Air Marshal Service to schedule air marshals on international flights) [136]. The audit game is a variation of the SSG designed to discover an efficient audit strategy [98, 99, 36, 37]. With respect to strategic auditing, most research has focused on deriving a defense strategy by solving, or approximating, the Strong Stackelberg Equilibrium (SSE). Unfortunately, it was recently shown that merely applying the SSE strategy may have limited efficacy in some security settings [107]. This can be addressed by strategically revealing information to the attacker [107, 104], a mechanism referred to as *signaling* (or *persuasion* [137, 108]). In this setting, the goal is to set up a *signaling scheme* to reveal noisy information to the attacker and, by doing so, influence the attacker's decision with respect to outcomes that favors the defender. However, all approaches derived to date rely on allocating resources

before signaling, such that it serves as a source of informational advantages for deceiving the attacker. Yet, in the audit setting, the decision sequence is reversed, such that the signal is revealed (e.g., via a warning screen) at the time of an access request, whereas the audit occurs after a certain period of time. This poses new challenges for the design of signaling schemes.

Many organizations have recognized and adopted signaling mechanisms to protect sensitive data. For example, in 2018, Vanderbilt University Medical Center (VUMC) announced a new *break-the-glass* policy to protect the privacy of patients with a *person of interest* (or VIP) designation, such as celebrities or public figures.[1] Under this policy, access to the EHRs of these individuals triggers a pop-up warning that requires the user to provide a justification for the access. Once the warning has been served, the user can decide whether or not to proceed to access, knowing that each access is logged for potential auditing. However, such a policy is implemented in a *post hoc* manner that does not optimize when to signal nor when to audit.

In this chapter, we introduce the notion of a Signaling Audit Game (SAG), which applies signaling to alert and auditing. We leverage the time gap between the access request made by the (potential) attacker and the actual execution of the attack to insert the signaling mechanism. When an alert is triggered by a suspicious access request, the system can, in real time, send a warning to the requestor. At this point, the attacker has an opportunity to re-evaluate his/her utility and make a decision about whether or not to continue with an attack. In contrast to previous models, which are all computed offline, the SAG optimizes both the warning strategy and the audit decision in real time for each incoming alert. Importantly, we consider the usability cost into the SAG where the normal data requestors may be scared away by the warning messages in practice. This may lead to descent in operational efficiency of organizations which deploy SAGs.

To illustrate the performance of the SAG, in this chapter we evaluate the expected utility

[1] https://www.mc.vanderbilt.edu/myvumc/index.html?article=21557

of the auditor with a dataset of over 10 million real VUMC EHR accesses and predefined alert types. The results of a comprehensive comparison, which is performed over a range of conditions, indicate that the SAG consistently outperforms state-of-the-art game theoretic alternatives that lack signaling by achieving higher overall utility while inducing nominal increases in computational burden.

The remainder of this chapter is organized as follows. We first propose the SAG and introduce how it is played in the audit setting. Next, we analyze the theoretical properties of the SAG equilibria. The dataset, experiments, and results are then described in the evaluation section. We conclude this chapter by discussing limitations of our approaches as opportunities for future investigation.

4.2 Online Signaling in Audit Games

In this section, we describe the SAG model in the general context of information services. For illustrative purposes, we use healthcare auditing as a running example.

4.2.1 Motivating Domain

To provide efficient healthcare service, HCOs typically store and process each patient's clinical, demographic, and financial information in an EHR system. EHR users, such as physicians and other clinical staff, need to access patients' EHRs when providing healthcare services. The routine workflow can be summarized as three steps: 1) a user initiates a search for a patient's EHR by name and date of birth, then the system returns a list of patients (often based on a fuzzy matching) along with their demographic information, 2) from the list, this user requests access to a patient's record, and 3) the system returns the requested record. Due to the complex, dynamic and time-sensitive nature of healthcare, HCOs typically grant employees broad access privileges, which unfortunately creates an opportunity for malicious insiders to exploit patients' EHRs [138].

To deter malicious access, breach detection tools are commonly deployed to trigger

alerts in real time for the administrator whenever suspicious events occur. Alerts are often marked with predefined types of potential violations which help streamline inspection. Notable alert types include accessing the EHR of co-workers, neighbors, family members, and VIPs [129]. Subsequently, a subset of the alerts are retrospectively audited at the end of each audit cycle, and the auditor determines which constitute an actual policy violation.

4.2.2 Signaling Audit Games

Here, we formalize the Signaling Auditing Game (SAG) model. An SAG is played between an *auditor* and an *attacker* within a predefined audit cycle (e.g., one day). This game is sequential such that alerts arrive one at a time. For each alert, the auditor needs to make two decisions in *real time*: first, which signal to send (e.g., to warn the user/attacker or not), and second, whether to audit the alert. Formally, let X_c^τ denote the event that alert τ will be audited, and X_u^τ denote that it is not audited. Following the convention of notations, the subscripts c and u stand for *covered* and *uncovered*, respectively. We further let ξ_1^τ denote the event that a *warning signal* is sent for alert τ, while ξ_0^τ denotes the event that no warning is sent (i.e. a "silent signal"). The warning ξ_1^τ is delivered privately through a dialog box on the requestor's screen, which might communicate "*Your access may be investigated. Would you like to proceed?*". $X_c^\tau, X_u^\tau, \xi_1^\tau, \xi_1^\tau$ are random variables whose probabilities are to be designated.

We assume that there is a finite set of alert types T and, for each $t \in T$, all alerts are considered equivalent for our purposes (i.e., attacks triggering alerts of type t all result in the same damages to the system). The auditor has an auditing budget B that limits the number of alerts that can be audited at the end of the cycle. For each alert type t, let V^t denote the cost (or time needed) to audit an alert of type t. Thus, if θ^t is the probability of auditing alerts of type t and d^t is the number of such alerts, the budget constraint implies that $\sum_t \theta^t \cdot V^t d^t \leq B$.

Since the setting is online, an optimal policy for the auditor must consider all possible

histories of alerts, including the correlation between alerts. Given that this is impractical, we simplify the scheme so that 1) each alert is viewed independently of alerts that precede it and 2) future alerts are considered with respect to their average relative frequency. Specifically, we assume that each attack effectively selects an alert type t, but do not need to consider the timing of attacks. Rather, we treat each alert as potentially adversarial. This implicitly assumes that an attack (e.g., a physician's access to the EHR of a patient they do not treat) triggers a single alert. However, this is without loss of generality, since we can define alert types that capture all realistic multi-alert combinations.

Now, we define the payoffs to the auditor and attacker. For convenience, we refer to the alert corresponding to an attack as the *victim alert*. If the auditor fails to audit a victim alert of type t, the auditor and the attacker will receive utility $U^t_{d,u}$ and $U^t_{a,u}$, respectively. On the other hand, if the auditor audits a victim alert of type t, the auditor and the attacker will receive utility $U^t_{d,c}$ and $U^t_{a,c}$, respectively. Here, the subscripts d and a stand for *defender* and *attacker*, respectively. Naturally, we assume $U^t_{a,c} < 0 < U^t_{a,u}$ and $U^t_{d,c} \geq 0 > U^t_{d,u}$.

Figure 4.1 demonstrates the key interactions of both players along the timeline. Each yellow block within the audit cycle represents a triggered alert and the corresponding interactions with it. The auditor continues to update the real time probability of auditing any alert (may or may not be triggered) with respect to the alert type and the time point τ. In other words, the auditor commits in real time to the auditing and signaling strategy. In this case, the auditor always moves first, as shown at the beginning of the lower timeline.

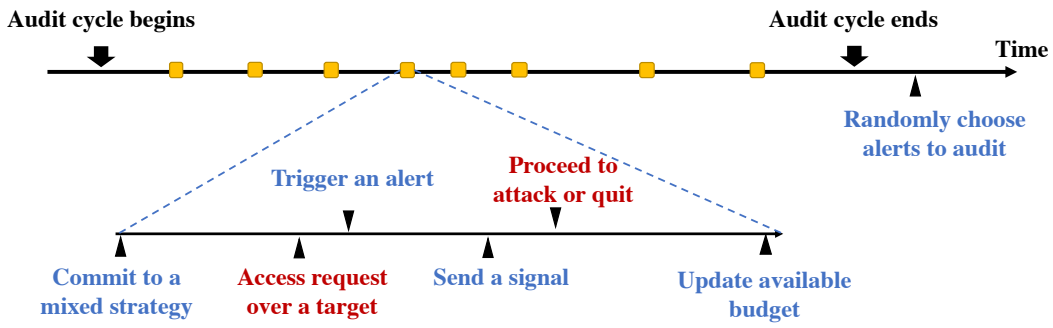


Figure 4.1: The auditor and attacker actions are shown in *blue* and *red*, respectively.

A *warning signaling scheme*, captured by the joint probability distribution of signaling and auditing, can be fully specified through four variables for each τ:

$$\begin{aligned} \mathbf{P}(\xi_1^\tau, X_c^\tau) = p_1^\tau, \quad \mathbf{P}(\xi_1^\tau, X_u^\tau) = q_1^\tau, \\ \mathbf{P}(\xi_0^\tau, X_c^\tau) = p_0^\tau, \quad \mathbf{P}(\xi_0^\tau, X_u^\tau) = q_0^\tau. \end{aligned} \tag{4.1}$$

Upon receiving the signal, the attacker reacts as follows:

- After ξ_1^τ: the system presents two choices to the attacker: *"Proceed"* to access the requested record or quit.
- After ξ_0^τ: the attacker automatically *proceeds* to access the requested record (since the attacker receives no warning).

For convenience, when possible we omit the superscript τ when the alert we are dealing with, is readily apparent from the context.

Figure 4.2 illustrates the temporal sequence of decisions in the SAG. Each edge in the figure is marked with its corresponding joint probability of a sequence of decisions up to and including that edge. Note that the two gray nodes are not extended because they do not lead to any subsequent event.[2]

Further, observe that, $p_1 + q_1 + p_0 + q_0 = 1$, and the overall probability of auditing this alert is $\mathbf{P}(X_c) = \mathbf{P}(X_c, \xi_1) + \mathbf{P}(X_c, \xi_0) = p_1 + p_0$. Conditional on the warning signal ξ_1, the probability of auditing this alert is thus $\mathbf{P}(X_c|\xi_1) = p_1/(p_1 + q_1)$.

Since the auditor has a fixed auditing budget, she will need to update the remaining budget after determining the signal-conditional audit probability for the current alert. We use B_τ to denote the remaining budget *before* receiving alert τ. Let t denote the type of alert τ and $\tau + 1$ denote the next alert. After the signaling scheme for τ is executed, the auditor then updates B_τ for the use of the next alert $\tau + 1$ as follows:

[2]The upper gray node corresponds to the case when an access request is abandoned. The lower one represents an impossible case because the user automatically gets the requested record.

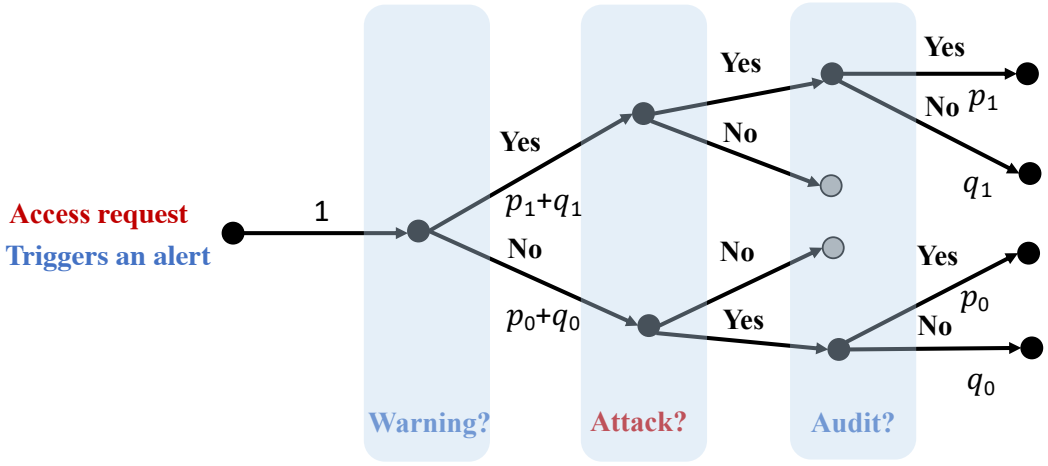


Figure 4.2: The decision tree of the auditor and an arbitrary user, the actions for which are shown in *blue* and *red*, respectively.

- If ξ_1^τ is sampled: $B_{\tau+1} = B_\tau - p_1^\tau/(p_1^\tau + q_1^\tau) \cdot V^t$.
- If ξ_0^τ is sampled: $B_{\tau+1} = B_\tau - p_0^\tau/(p_0^\tau + q_0^\tau) \cdot V^t$.

Additionally, we always ensure that $B_\tau \geq 0$. The key challenge in our model is to compute the optimal $p_1^\tau, q_1^\tau, p_0^\tau, q_0^\tau$ for each alert τ *online* by accounting for the remaining budget and the estimate number of future alerts. This needs to be performed to ensure that the auditor does not spend the budget at a rate that is excessively fast or slow.

Without signaling, our audit game can be solved *offline*, at the end of the audit cycle. This situation can be captured by a Stackelberg security game by viewing alerts as targets. The optimal auditing probabilities can then be determined offline by computing the SSE of this game. However, as our experiments show, this simplified strategy (which we refer to as *offline SSE*) performs substantially worse than our online approach.

The SAG can be viewed as a variation on the Stackelberg game, where it includes signaling and makes decisions about auditing *online* upon the arrival of each alert. The premise behind our solution is therefore a Strong Stackelberg equilibrium of the SAG, in which the auditor commits to a randomized joint signaling and auditing decision, and the associated probability distribution is observed by the attacker, who then decides first upon

the alert type to use, and subsequently whether to proceed after a warning. We will seek the optimal randomized commitment strategy for the auditor in this game.

The SAG model contains two crucial differences from prior investigations into signaling for security games. The first is that the signaling scheme for each alert in an SAG must be optimized sequentially in *real time*. By contrast, previous models, such as [107], decide the signaling schemes for all targets simultaneously in an offline fashion. The second is in how private information is leveraged. In previous models, the defender utilizes the informational advantage that the defender *currently* has (e.g., knowledge about the realized protection status of the target) to deceive the attacker. However, in our scenario, the auditor first decides the signaling scheme, by when he/she has an equal amount of information as the attacker (which includes the status of the current environment), and then exercises her informational advantage *after* the audit cycle ends (by deciding which to audit).

4.3 Optimizing SAGs

In this section, we design an algorithm for solving SAGs. For presentation purpose, we fix the alert τ to a particular type t and, thus, the superscript will, at times, be omitted for notational convenience. We begin by considering the problem of computing the real time SSE of the game without signaling that transpires for a given observed alert τ. This game, as well as its solution, serve as a baseline of the optimized SAGs.

4.3.1 Online SSG

Consider the arrival of an alert τ. Let d_τ^t be the number of future alerts of type $t \in T$ after alert τ is triggered.[3] We assume that d_τ^t follows a Poisson distribution D_τ^t, which is widely adopted to characterize the number of arrivals. We can compute the SSE strategy using a multiple linear programming (LP) approach for budget B_τ. In this approach, for

[3]The vast majority of alerts are false positives. Consequently, we can estimate d_τ^t from alert logs in previous audit cycles.

each alert type t, we assume that t is the attacker's best response, and then compute the optimal auditing strategy. Finally, we choose the best solution (in terms of the auditor's utility) among all of the LPs as the SSE strategy.

Now, let $\theta^{t'}(t)$ be the probability of auditing an alert of type t' when the attacker's best response is t. In addition to this optimal auditing policy, we design how we plan to split the remaining budget B_τ among all alert types. We assume that the audit distribution will remain constant for future alerts, which allows us to consider the long-term impact of our decision about auditing. We represent the budget that we allocate for inspecting alerts of each type as a vector $B_\tau = \{B_\tau^1, B_\tau^2, ..., B_\tau^{|T|}\}$ that the long-term budget allocation decision is constrained by the remaining audit budget: $\sum_{t=1}^{|T|} B_\tau^t \leq B_\tau$. Now, assuming type t is the best response, the following LP returns optimal auditing strategy:

$$
\begin{aligned}
& \max_{B_\tau} \theta^t(t) \cdot U_{d,c}^t + (1-\theta^t(t)) \cdot U_{d,u}^t \\
s.t. & \\
& \forall t', \quad \theta^t(t) \cdot U_{a,c}^t + (1-\theta^t(t)) \cdot U_{a,u}^t \\
& \qquad\qquad \geq \theta^{t'}(t) \cdot U_{a,c}^{t'} + (1-\theta^{t'}(t)) \cdot U_{a,u}^{t'}, \\
& \forall t', \quad \theta^{t'}(t) = \mathbb{E}_{d_\tau^{t'} \sim D_\tau^{t'}} \left(\frac{B_\tau^{t'}}{V^{t'} d_\tau^{t'}} \right), \\
& \qquad \sum_{t'=1}^{|T|} B_\tau^{t'} \leq B_\tau, \\
& \forall t', \quad B_\tau^{t'} \in [0, B_\tau],
\end{aligned}
\tag{4.2}
$$

where the first constraint ensures that t is the attacker's best response. After solving $|T|$ instances of LP (4.2), the best solution for the auditor will henceforth be referred to as the *online SSE strategy* (or simply, the *SSE*), θ_{SSE}.

4.3.2 Optimal Signaling

We now describe how to build a signaling mechanism into the audit game and then compute the optimal signaling scheme, as well as the budget allocation strategy.

From the perspective of the attacker, whether to *proceed* or *quit* after receiving a warning signal depends on his conditional expected utility:

$$\mathbb{E}_a^t(util|\xi_1) = \frac{p_1^t}{p_1^t + q_1^t} \cdot U_{a,c}^t + \frac{q_1^t}{p_1^t + q_1^t} \cdot U_{a,u}^t.$$

We impose the constraint $\mathbb{E}_a^t(util|\xi_1) \leq 0$ such that the attacker's best response to ξ_1 is to quit, in which case both players will receive 0 utility. We do not enforce constraints for ξ_0 because the potential attacker does not have any option but to proceed. In this case, the expected utility of the auditor is

$$\mathbb{E}_d^t(util|\xi_0) = \frac{p_0^t}{p_0^t + q_0^t} \cdot U_{d,c}^t + \frac{q_0^t}{p_0^t + q_0^t} \cdot U_{d,u}^t.$$

Overall, the expected utility for the attacker can be computed as

$$\mathbb{E}_a^t(util) = (p_0^t + q_0^t) \cdot \mathbb{E}_a^t(util|\xi_0) = p_0^t \cdot U_{a,c}^t + q_0^t \cdot U_{a,u}^t.$$

Accordingly, the auditor's expected utility is

$$\mathbb{E}_d^t(util) = (p_0^t + q_0^t) \cdot \mathbb{E}_d^t(util|\xi_0) = p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t.$$

However, a side effect is that, the warnings sent by the auditor (e.g., the pop-up warning screen off of *break-the-glass* strategy deployed by VUMC) may pose an additional utility loss to the auditor in practice, which we call *usability cost*. This is because when normal users request access to sensitive data and receive a warning message, they may walk away by choosing quit instead of "Proceed", which induces a loss in operational efficiency for the organization. For each type t', we set this loss to be proportional to the product of the probability of sending warnings $p_1^{t'} + q_1^{t'}$, the probability of being deterred $P^{t'}$ and the expectation of the number of future false positive alerts to the end of the current audit

cycle $E_\tau^{t'}$. The loss incurred for each quit by a normal user is set to be $C_{t'}(<0)$. Then, the expected utility of the auditor can be updated as $\mathbb{E}_d^t(util) = p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t + \sum_{t'=1}^{|T|}(p_1^{t'} + q_1^{t'}) \cdot P^{t'} \cdot E_\tau^{t'} \cdot C_{t'}$.

The optimal signaling scheme (or, more concretely, joint signaling and audit probabilities) can be computed through the following set of LPs:

$$
\begin{aligned}
&\max_{p_0,p_1,q_0,q_1,B_\tau} p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t + \sum_{t'=1}^{|T|}(p_1^{t'} + q_1^{t'}) \cdot P^{t'} \cdot E_\tau^{t'} \cdot C_{t'} \\
&s.t. \\
&\quad \forall t', \quad p_0^t \cdot U_{a,c}^t + q_0^t \cdot U_{a,u}^t \geq p_0^{t'} \cdot U_{a,c}^{t'} + q_0^{t'} \cdot U_{a,u}^{t'}, \\
&\quad \forall t', \quad p_1^{t'} \cdot U_{a,c}^{t'} + q_1^{t'} \cdot U_{a,u}^{t'} \leq 0, \\
&\quad \forall t', \quad p_1^{t'} + p_0^{t'} = \mathbb{E}_{d_\tau^{t'} \sim D_\tau^{t'}} \left(\frac{B_\tau^{t'}}{V^{t'} d_\tau^{t'}} \right), \\
&\quad \forall t', \quad p_1^{t'} + p_0^{t'} + q_1^{t'} + q_0^{t'} = 1, \\
&\qquad \sum_{t' \in \{1,\ldots,|T|\}} B_\tau^{t'} \leq B_\tau, \\
&\quad \forall t', \quad B_\tau^{t'} \in [0, B_\tau], \\
&\quad \forall t', \quad p_0^{t'}, q_0^{t'}, p_1^{t'}, q_1^{t'} \in [0,1],
\end{aligned}
\tag{4.3}
$$

where we assume type t is the best one for the attacker to potentially exploit. Note that, in the objective function, the incurred additional loss is an accumulated value that considers the amount of time remaining in the period for the current audit cycle. The likelihood of sending warning signal in the current time point is a real time estimation of future warnings. Due to the fact that attacks are extremely rare in practice in comparison to the magnitude of alerts, in solving LP (4.3) we use the expected number of future alerts $\mathbb{E}_{d_\tau^{t'} \sim D_\tau^{t'}}(d_\tau^{t'})$ to approximate $E_\tau^{t'}$. As a result, $\mathbb{E}_{d_\tau^{t'} \sim D_\tau^{t'}}(d_\tau^{t'})$ can then be estimated from historical data collected in previous audit cycles. Our goal is thus to find the optimal signaling scheme for all types, and simultaneously, the best budget allocation strategy. We use p_0, p_1, q_0 and q_1 to denote the warning signaling scheme for all types, namely, the set $\{p_0^{t'} | \forall t'\}$, $\{p_1^{t'} | \forall t'\}$,

$\{q_0^{t'}|\forall t'\}$ and $\{q_1^{t'}|\forall t'\}$, respectively.

The first constraint in LP (4.3) ensures that attacking type t is the best response strategy for the attacker. The second constraint indicates that the attacker, when receiving a warning signal, will quit attacking any type. We refer to the optimal solution among the $|T|$ instances of LP (4.3) as the *Online Stackelberg Signaling Policy (OSSP)*. In particular, we use θ_{ossp} to denote the vector of coverage probability at OSSP.

After building the theoretical model of the SAG, we need to pay attention to one important situation in practice, where an attacker can leverage to perform attacks with lower level risks of being captured.

4.3.3 The Ending Period of Audit Cycles

Recall that in SAGs, the estimation of the number of alerts in the rest of the current audit cycle, which is $\mathbb{E}_{d_\tau^t \sim D_\tau^t}(d_\tau^t)$, is calculated based on the alert logs of historical audit cycles. At the ending period of audit cycles, such estimation keeps decreasing for each type. As a consequence, it would be ill-advised to apply any approach that performs an estimation on the arrivals without an additional process to handle the ending period of an audit cycle. Imagine, for instance, an attacker who only attacks at the very end of an audit cycle. Then, the knowledge from historical data is likely to indicate that no alerts will be realized in the future. And it follows that such attacks will not be covered because the available budget will have been exhausted according to the historical information.

To practically mitigate this problem, when the mean of arrivals in the historical data drops under a certain threshold, we apply the estimate of the number of future alerts $\mathbb{E}_{d_{\tau-1}^t \sim D_{\tau-1}^t}(d_{\tau-1}^t)$ in the time point when the last alert was triggered as a proxy of the real one at the current time point. This technique is called *knowledge rollback*. By doing so, the consumption of the available budget in real time will be slowed down because of the application of a smaller coverage probability. As a consequence, the attacker attempting to attack late is not afforded an obvious extra benefit.

4.4 Theoretical Properties of SAGS

In this section, we theoretically analyze the properties of the OSSP solution (equivalently, of the SAG equilibrium). Our first result highlights a notable property of the optimal signaling scheme. Specifically, the optimal signaling scheme will only trigger warning signals for the best attacking type, i.e., the type at which attacker utility is maximized. As such, the rational attacker will choose to attack this alert type.

Theorem 2 *If alert τ_* of type t_* is the best response strategy for the attacker, then $p_1^t = q_1^t = 0$ in the OSSP for $\forall t \neq t_*$.*[4]

Proof Let $Sol = \{p_0^t, p_1^t, q_0^t, q_1^t\}_{t \in T}$ be any optimal solution and t_* is the best type. We show that the following newly defined variables will not decrease the objective value of *Sol* and thus, by assumption, is still optimal. Let $\bar{p}_0^{t_*} = p_0^{t_*}, \bar{p}_1^{t_*} = p_1^{t_*}, \bar{q}_0^{t_*} = q_0^{t_*}, \bar{q}_0^{t_*} = q_0^{t_*}$ be the same as in *Sol*, however for any $t \neq t_*$, define $\bar{p}_0^t = p_0^t + p_1^t, \bar{q}_0^t = q_0^t + q_1^t$ and $\bar{p}_1^t = 0, \bar{q}_1^t = 0$.

First, we argue that these newly defined variables are still feasible. All of the constraints can easily be verified in LP (4.3) except the first two sets. The second set of constraints is still satisfied for any $t \neq t_*$ (where our variables changed) since $\bar{p}_1^t = \bar{q}_1^t = 0$. The first set of constraints are satisfied for any $t \neq t_*$ because

$$\begin{aligned} \bar{p}_0^t \cdot U_{a,c}^t + \bar{q}_0^t \cdot U_{a,u}^t &= (p_0^t + p_1^t) \cdot U_{a,c}^t + (q_0^t + q_1^t) \cdot U_{a,u}^t \\ &\leq p_0^{t_*} \cdot U_{a,c}^t + q_0^{t_*} \cdot U_{a,u}^t \\ &= \bar{p}_0^{t_*} U_{a,c}^t + \bar{q}_0^{t_*} U_{a,u}^{t_*}, \end{aligned}$$

where the (only) inequality is due to $p_1^t \cdot U_{a,c}^t + q_1^t \cdot U_{a,u}^t \leq 0$ as a constraint of LP (4.3) and the two equations are by our definition of the new variables. This proves that the first constraint is also feasible.

[4] We will use $*$ to denote strategies or quantities in the OSSP in the rest of the chapter.

It remains to show that the newly defined variables do not decrease the objective function. This follows simply because the term with respect to type t_* in the objective function does not change and all the other terms become zero in the newly defined variables, which is no less than the original cost. This proves the theorem. ■

Theorem 2 leads to the following corollary: when the attacker avoids attacking certain type(s) at any time point (this is always the case in OSSP), then the best strategy for the auditor is to turn off the signaling procedure for those types for less loss incurred by sending warnings. Now we show that, at any given game status, the marginal coverage probability for OSSP is the same as the one for the online SSE.

Theorem 3 *Let θ^t_{ossp} be the marginal coverage probability in the OSSP at any given game status and θ^t_{SSE} be the corresponding marginal coverage probability in the online SSE. Then, in a SAG, for each type $t \in T$, $\theta^t_{ossp} = \theta^t_{SSE}$.*

Proof Given any game state, the auditor has an estimate about the sets of future alerts. We prove that for any fixed set of alerts, $\theta^t_{ossp} = \theta^t_{SSE}$ holds for each type $t \in T$. As a result, in expectation over the probabilistic estimate, this still holds.

Fixing a set of alerts, the auditor's decision is a standard Stackelberg game. We first claim that by fixing the auditing strategy in the OSSP, the attacker can receive $\mathbb{E}_a^{ossp}$ by triggering any alert τ, thus type t. In other words, $\forall t \neq t_*, \mathbb{E}_a(\theta^t_{ossp}) = \mathbb{E}_a^{ossp}$. Assume, for the sake of contradiction, that an alert τ' of type t' with positive coverage probability is not the best response of the attacker in an SAG. Then, the auditor can redistribute a certain amount of the protection resources from τ' to the alerts of the attacker's best-response type and guarantee that it is still the best-response type. This increases the coverage probability of these alerts and, thus, increases the auditor's utility, which contradicts the optimality of OSSP. This implies that the first constraint in LP (4.3) is *tight* in the OSSP. Similarly, this holds true for the online SSE. Notice that $\mathbb{E}_a(\theta^t)$ is a strictly decreasing function of θ^t for both OSSP and online SSE.

Next, we prove that $\mathbb{E}_a^{sse} = \mathbb{E}_a^{ossp}$ implies $\theta_{ossp}^t = \theta_{SSE}^t$ for all τ, thus t, as desired. This is because $\theta_{ossp}^t > \theta_{SSE}^t (\geq 0)$ implies $\mathbb{E}_a^{ossp} = \mathbb{E}_a(\theta_{ossp}^t) < \mathbb{E}_a(\theta_{SSE}^t) = \mathbb{E}_a^{sse}$ (a contradiction) and $\theta_{ossp}^t < \theta_{SSE}^t$ implies $\mathbb{E}_a^{ossp} \geq \mathbb{E}_a(\theta_{ossp}^t) > \mathbb{E}_a(\theta_{SSE}^t) = \mathbb{E}_a^{sse}$ (again, a contradiction). As a result, it must be the case that $\theta_{SSE}^t = \theta_{ossp}^t$ for all τ, and thus t, as desired.

We now show that $\mathbb{E}_a^{sse} = \mathbb{E}_a^{ossp}$ must hold true. Assume, for the sake of contradiction, that $\mathbb{E}_a^{sse} > \mathbb{E}_a^{ossp}$. Then for any $\theta_{SSE}^t > 0$, it must be that $\theta_{ossp}^t > \theta_{SSE}^t$. This is because $\theta_{ossp}^t \leq \theta_{SSE}^t$ implies that $\mathbb{E}_a^{ossp} \geq \mathbb{E}_a(\theta_{ossp}^t) \geq \mathbb{E}_a(\theta_{SSE}^t) = \mathbb{E}_a^{sse}$, which is a contradiction. On the other hand, for any $\theta_{ossp}^t > 0$, $\theta_{SAG}^t > \theta_{SSE}^t$ must be true, because $0 < \theta_{ossp}^t \leq \theta_{SSE}^t$ implies that $\mathbb{E}_a^{sse} = \mathbb{E}_a(\theta_{SSE}^t) \leq \mathbb{E}_a(\theta_{ossp}^t) = \mathbb{E}_a^{ossp}$, which is a contradiction. As a result, it must be the case that either $\theta_{SSE}^t = \theta_{ossp}^t = 0$ or $\theta_{ossp}^t > \theta_{SSE}^t$ for any τ, thus t. Yet this contradicts the fact that $\sum_\tau \theta_{SSE}^t = \sum_\tau \theta_{ossp}^t = B_\tau$. Similarly, $\mathbb{E}_a^{sse} < \mathbb{E}_a^{ossp}$ can not hold true. As a result, $\mathbb{E}_a^{sse} = \mathbb{E}_a^{ossp}$ is true. ■

In the proof above, we can conclude that the attacker's utility is the same in the OSSP and the online SSE. We now prove that the SAG is lower-bounded by the online SSG with respect to the auditor's expected utility.

Theorem 4 *Given any game state, the expected utility of the auditor by applying the OSSP is never worse than when the online SSE is applied.*

Proof If the attacker completes the attack, his expected utility by attacking type t in SAG is $\mathbb{E}_a(\theta^t) = (p_1^t + p_0^t) \cdot U_{a,c}^t + (q_1^t + q_0^t) \cdot U_{a,u}^t$, where θ^t is the coverage probability of type t.

- If $\mathbb{E}_a(\theta^t) < 0$, then the attacker will choose to not approach any target at the beginning, regardless of if there exists a signaling procedure. Thus, in both cases the auditor will achieve the same expected utility, which is 0.

- If $\mathbb{E}_a(\theta^t) \geq 0$, then let $p_1^t = 0$ and $q_1^t = 0$. And it follows that $p_0^t = \theta^t$ and $q_0^t = 1 - \theta^t$. This solution satisfies all of the constraints in LP (4.3), which, in this case, share exactly the same form with LP (4.2). In combination with Theorem 2, we can

conclude that in this special setting, the expected utilities of the auditor, by applying SAG (not necessary the OSSP) and online SSE, are the same: $\mathbb{E}_d(\theta^t) = \theta^t \cdot U^t_{d,c} + (1-\theta^t) \cdot U^t_{d,u}$. Thus, the expected utility of the auditor in the OSSP is never worse than the one in the online SSE. ■

This begs the following question: can applying the OSSP bring more benefit to the expected utility of the auditor? Our experiments lend support to an affirmative answer.

Our next result reveals an interesting property about the optimal signaling scheme. Interestingly, it turns out that by applying OSSP in specific situations, if there is no warning sent, then the auditor will not audit the triggered alerts in their optimal strategy (i.e., $p^{t_*}_0 = 0$).

Theorem 5 *In SAG, if the payoff structure satisfies* $0 \geq (U^{t_*}_{d,c} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*})/(U^{t_*}_{d,u} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}) \geq U^{t_*}_{a,c}/U^{t_*}_{a,u}$ *on the best attacking type* t_* *in the OSSP, then we have* $p^{t_*}_0 = 0$ *on the* τ*–th alert.*

Proof This will be proved in the instance of LP (4.3) that derives the best pair of the signaling strategy and the attacking strategy t_*. For inference convenience, for all t we substitute p^t_1 and q^t_1 with $\theta^t_{ossp} - p^t_0$ and $1 - \theta^t_{ossp} - q^t_0$, respectively. Combining with Theorem 2, the objective function of LP (4.3) can be simplified as $p^{t_*}_0 \cdot U^{t_*}_{d,c} + q^{t_*}_0 \cdot U^{t_*}_{d,u} + p^{t_*}_1 \cdot P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*} + q^{t_*}_1 \cdot P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*} = p^{t_*}_0 \cdot (U^{t_*}_{d,c} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}) + q^{t_*}_0 \cdot (U^{t_*}_{d,u} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}) + P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}$.

Now, we simplify constraints. The first constraint is always tight in the OSSP (as shown in Theorem 3). By applying the substitution rules, the second constraint becomes $\forall t',\ p^{t'}_0 \cdot U^{t'}_{a,c} + q^{t'}_0 \cdot U^{t'}_{a,u} \geq \theta^{t'}_{ossp} \cdot U^{t'}_{a,c} + (1 - \theta^{t'}_{ossp}) \cdot U^{t'}_{a,u}$. For all $t' \neq t_*$, it can be future transformed into $(\theta^{t'}_{ossp} - p^{t'}_0) \cdot U^{t'}_{a,c} + (1 - \theta^{t'}_{ossp} - q^{t'}_0) \cdot U^{t'}_{a,u} \leq 0$. Due to the fact that $p^{t'}_1 = q^{t'}_1 = 0$ in the OSSP for $\forall t' \neq t_*$, $p^{t'}_0$ is equal to $\theta^{t'}_{ossp}$, and $q^{t'}_0$ equal to $1 - \theta^{t'}_{ossp}$. As such, for all $t' \neq t_*$, this constraint naturally holds true. By far, the best strategy pair of SAG in our setting needs to maximize $p^{t_*}_0 \cdot (U^{t_*}_{d,c} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}) + q^{t_*}_0 \cdot (U^{t_*}_{d,u} - P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}) + P^{t_*} \cdot E^{t_*}_{\tau} \cdot C_{t_*}$, such that $p^{t_*}_0 \cdot U^{t_*}_{a,c} + q^{t_*}_0 \cdot U^{t_*}_{a,u} \geq \theta^{t_*}_{ossp} \cdot U^{t_*}_{a,c} + (1 - \theta^{t_*}_{ossp}) \cdot U^{t_*}_{a,u}$ (we refer to this inequality as constraint

α) and that these probability variables are in $[0,1]$ and sum up to 1.[5]

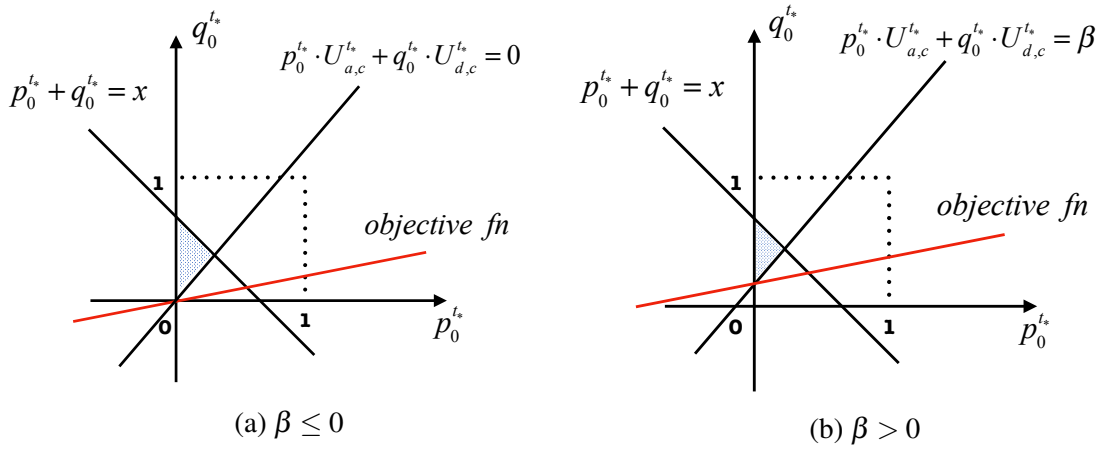
We set up a Cartesian coordinate system and let $q_0^{t_*}$ be the vertical axis and $p_0^{t_*}$ the horizontal one. Geometrically, the slopes of the item to be maximized, which is $-(U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*})/(U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*})$ and constraint α, which is $-U_{a,c}^{t_*}/U_{a,u}^{t_*}$ are both positive. Note that, though we do not constrain the left side of constraint α, which is $\mathbb{E}_a^{t_*}(util|\xi_u) = p_0^{t_*} \cdot U_{a,c}^{t_*} + q_0^{t_*} \cdot U_{a,u}^{t_*} > 0$, this inequality is always true. If not the case, the attacker will not initially attack. We discuss the righthand side $\beta = \theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*}$ as follows.

- $\beta \leq 0$. In this setting, constraint α is dominated. The boundary of the dominant constraint passes the origin and the feasible region is a triangle with its base on the vertical axis, as shown in Figure 4.3a. Thus, in both cases, if $(U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*})/(U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) \geq U_{a,c}^{t_*}/U_{a,u}^{t_*}$ holds true (which implies that the slope of the objective function is less than the boundary's slope of the dominant constraint), then $p_0^{t_*} = q_0^{t_*} = 0$ leads to the maximum of the objective function. The OSSP, thus is $p_1^{t_*} = \theta_{ossp}^{t_*}, q_1^{t_*} = 1 - \theta_{ossp}^{t_*}, p_0^{t_*} = q_0^{t_*} = 0$.

- $\beta > 0$. Thus, constraint α dominates $p_0^{t_*} \cdot U_{a,c}^{t_*} + q_0^{t_*} \cdot U_{a,u}^{t_*} > 0$. The boundary's intercept of the dominant constraint is $\delta = (\theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*})/U_{a,u}^{t_*} \in (0,1]$. Using an analysis similar to the previous case of β, only when $p_0^{t_*} = 0, q_0^{t_*} = \delta$ does lead to the maximum of the objective function. This is indicated in Figure 4.3b. The OSSP is $p_1^{t_*} = \theta_{ossp}^{t_*}, p_0^{t_*} = 0, q_1^{t_*} = 1 - \theta_{ossp}^{t_*} - (\theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*})/U_{a,u}^{t_*}, q_0^{t_*} = (\theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*})/U_{a,u}^{t_*}$. ■

Remark. In application domains, the absolute value of the penalty for the attacker is often greater than the benefit from committing attacks. As for the auditor, his/her benefit from catching an attack is often less than the absolute value of the loss due to missing an attack. If the warning cost $P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$ were ignored, then $0 \geq U_{d,c}^{t_*}/U_{d,u}^{t_*} \geq U_{a,c}^{t_*}/U_{a,u}^{t_*}$ is often satisfied in practice. Considering that the warning cost is proportional to the estimation of

[5] Constraints involving $B_\tau^{t'}$ are neglected because $\theta_{ossp}^{t'}$ is the coverage probability that can be derived from $B_\tau^{t'}$ in our setting.



(a) $\beta \leq 0$ (b) $\beta > 0$

Figure 4.3: Feasible regions (blue areas) and an objective function gaining the largest value for $\beta \leq 0$ and $\beta > 0$. Note that the boundary $p_0^{t_*} + q_0^{t_*} = x$ is only for illustration, and its intercept can slide in $[0,1]$ by taking into account the value of $p_1^{t_*}$ and $p_1^{t_*}$. However, this never impact the optimal solution.

the number of future warning events, which decreases with time, the condition in Theorem 5 only happens in a certain period of time.

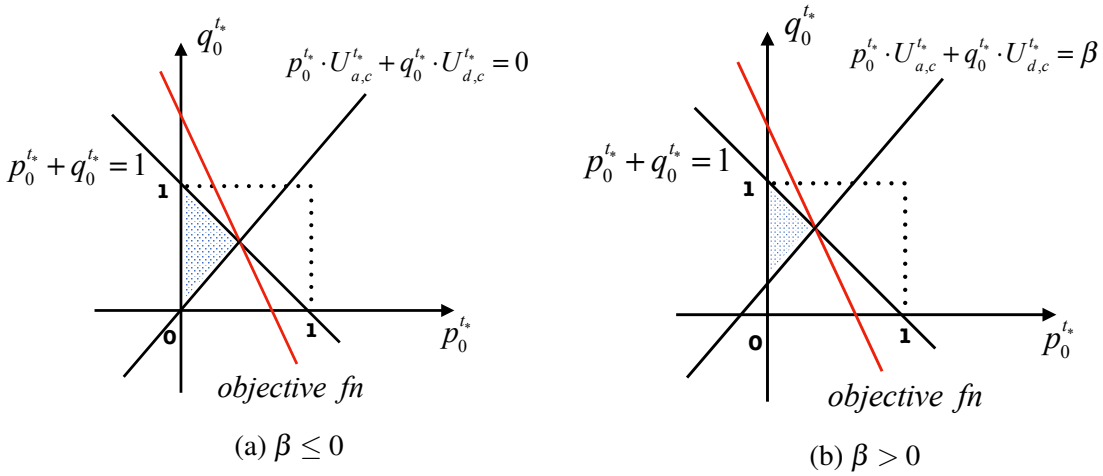
One might wonder that, given that the condition in Theorem 5 is valid, whether the attacker can keep attacking until receiving no warning, in which case the attacker can attack safely under the optimal signaling scheme? Actually, this strategy cannot lead to success because once the attacker chooses to quit, his/her identity is essentially revealed. The auditor cannot punish the attacker (yet) because the attacker quits the attack, leaving no evidence. Therefore, a successful attack later on only hurts him/her, while help the auditor find forensic evidence of an attack. In practice, it is common that the auditor uses reserved budget to deal with special cases. In the setting above, the author can use a small portion of the auditing budget to investigate repeated attempts of data access, but in practice this is not an issue, as these cases are likely to be rare in real world. As a result, once an attacker chooses to quit, the best response should be to not attack during the rest of the auditing cycle. In the experimental comparison with online/offline SSG, which requires no additional budget for such attack category, we will apply a reduced available budget as the input of the corresponding SAG to ensure fairness in our comparisons.

A natural follow-up question is can the attacker manipulate the model by running this strategy across audit cycles? The answer is no as well. Such a behavior can be easily detected by a rule that applies when the attacker performs his/her attack repeatedly. When the auditor does not send a warning, the attacker successfully attacks. Yet, since there was a warning sent previously, the auditor will use the probability p_1 to audit, rather than p_0. Thus, the attacker should take this into account before adopting such a strategy.

Theorem 6 *The auditor benefits equally in terms of the expected utility from SAG and online SSG at the τ-th alert, if it satisfies $U_{d,u}^{t_*} > P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$, where t_* is the best type to attack in the OSSP.*

Proof We prove this by applying the same simplification and the split strategy (i.e., analyze two distinct situations based on the value of β) as applied in the proof for Theorem 5. Note that the slope of the objective function is $-(U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*})/(U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*})$. Since $C_{t_*} < 0$, the numerator is less than 0. If $U_{d,u}^{t_*} > P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$, then the denominator is greater than 0. Thus, the slope is less than 0. In particular, the slope is less than -1 (which is the slope of boundary $p_0^{t_*} + q_0^{t_*} = x$) because of $U_{d,c}^{t_*} \geq 0 > U_{d,u}^{t_*}$. We now analyze properties in this situation geometrically.

As demonstrated in Figures 4.4a and 4.4b, the boundary $p_0^{t_*} + q_0^{t_*} = x (\in [0,1])$ should pass through the $(0,1)$ point. This is because, if this failed to occur, then the value of the objective function can be further improved by lifting the boundary. The optimal solution for both cases is at the intersection point of the two boundaries of the feasible region. Thus, it follows that $p_0^{t_*} + q_0^{t_*} = 1$ for the OSSP, which implies $p_1^{t_*} = q_1^{t_*} = 0$. In other words, the signaling procedure is turned off for the best attacking type t_* in the OSSP. Combining with what Theorem 2 indicates, when $U_{d,u}^{t_*} > P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$, the signaling procedure is off for all types. In LP (4.3), by substituting variables $p_1^{t'}$ and $q_1^{t'}$ (for all t') with 0, the SAG instance becomes an online SSG (as shown in LP (4.2)). Thus, the two LPs share the same solution, and the auditor will receive the same expected utility in both auditing mechanism. ■



(a) $\beta \leq 0$ (b) $\beta > 0$

Figure 4.4: Feasible regions (blue shaded triangle areas) and an objective function gaining the largest value for $\beta \leq 0$ and $\beta > 0$.

This result indicates that if the incurred loss due to a warning is too large, then an SAG will degrade into an online SSG, where the signaling procedure is turned off. It suggests that in the application domain, to ensure that the signaling is deployed in a useful manner, organizations need to 1) refine the alert system so that false positive alerts can be classified as normal events, and 2) decrease the number of the events in which normal users are scared away.

Our final theory concerns the attacker's utility in OSSP.

Theorem 7 *The expected utility of the attacker when applying the OSSP is the same as that achieved when applying the online SSE strategy.*

Proof Let $\mathbb{E}_a^{sse}$ and $\mathbb{E}_a^{ossp}$ denote the expected utility for the attacker at online SSE and OSSP, respectively. Assume, for the sake of contradiction, that $\mathbb{E}_a^{sse} > \mathbb{E}_a^{ossp}$. Then for any $\theta_{SSE}^t > 0$, we must have $\theta_{ossp}^t > \theta_{SSE}^t$. This is because $\theta_{ossp}^t \leq \theta_{SSE}^t$ implies that $\mathbb{E}_a^{ossp} \geq \mathbb{E}_a(\theta_{ossp}^t) \geq \mathbb{E}_a(\theta_{SSE}^t) = \mathbb{E}_a^{sse}$, which is a contradiction. On the other hand, for any $\theta_{ossp}^t > 0$, $\theta_{SAG}^t > \theta_{SSE}^t$ must be true, because $0 < \theta_{ossp}^t \leq \theta_{SSE}^t$ implies that $\mathbb{E}_a^{sse} = \mathbb{E}_a(\theta_{SSE}^t) \leq \mathbb{E}_a(\theta_{ossp}^t) = \mathbb{E}_a^{ossp}$, which is a contradiction. As a result, it must be the case that either $\theta_{SSE}^t = \theta_{ossp}^t = 0$ or $\theta_{ossp}^t > \theta_{SSE}^t$ for any τ, thus t. Yet this contradicts the fact that $\sum_\tau \theta_{SSE}^t =$

Table 4.1: A summary of the daily statistics per alert types.

ID	Alert Type Description	Mean	Std
1	Same Last Name	196.57	17.30
2	Department Co-worker	29.02	5.56
3	Neighbor ($\leq$ 0.5 miles)	140.46	23.23
4	Same Address	10.84	3.73
5	Last Name; Neighbor ($\leq$ 0.5 miles)	25.43	4.51
6	Last Name; Same Address	15.14	4.10
7	Last Name; Same Address; Neighbor ($\leq$ 0.5 miles)	43.27	6.45

$\sum_\tau \theta^t_{ossp} = B_\tau$. Similarly, $\mathbb{E}^{sse}_a < \mathbb{E}^{ossp}_a$ can not hold true. As a result, $\mathbb{E}^{sse}_a = \mathbb{E}^{ossp}_a$ is true. ■

4.5 Model Evaluation

In this section, we evaluate the performance of the SAG on the real EHR access logs from VUMC, which deployed an unoptimized warning strategy. To illustrate the value of signaling, we compare with multiple game theoretic alternative methods in terms of the expected utility of the auditor. Specifically, we investigate the robustness of the advantage of SAGs under a range of different conditions. Now we first describe the real dataset which is used for evaluation.

Table 4.2: The payoff structures for the pre-defined alert types.

Payoff	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7
$U_{d,c}$	100	150	150	300	400	600	700
$U_{d,u}$	-400	-500	-600	-800	-1000	-1500	-2000
$U_{a,c}$	-2000	-2250	-2500	-2500	-3000	-5000	-6000
$U_{a,u}$	400	400	450	600	650	700	800

4.5.1 Dataset

The dataset consists of EHR access logs for 56 continuous normal working days in 2017. We excluded all holidays (include weekends) because they exhibit a different access pattern from working days. The total number of unique accesses $\langle$Date, Employee, Patient$\rangle$ is on the order of $10.75M$. The mean and standard deviation of daily unique accesses

are approximately $192K$ and $8.97K$, respectively. We focus on the following alerts types: employee and patient: 1) share the same last name, 2) work in the same department, 3) share the same residential address, and 4) are neighbors within a distance less than 0.5 miles. When an access triggers multiple distinct types of alerts, their combination is regarded as a new type. Table 5.1 lists the set of predefined alert types, along with the mean and standard deviation of their occurrence on a daily basis. We provide the payoff structure for both the attacker and the auditor in Table 5.2. These values are estimates based on discussions with experts working in the area.

4.5.2 Experimental Setup

The audit cycle is defined as one day from *0:00:00* to *23:59:59*. From the dataset, we construct 15 groups, each of which contains the alert logs of 41 continuous normal working days as the historical data (for estimating the distributions of future alerts in all types), and the alert logs of the 1 subsequent day as the day for testing purpose. We set up a real time environment for evaluating the performance in terms of the auditor's expected utility. We set the audit cost per alert to $V^t = 1, \forall t \in \{1, ..., |T|\}$. From the alert logs of three months, we obtain the frequency at which users quit when they receive the warning messages in our dataset. According to this observation, in our experiments we set the probability of quitting as $P^t = 0.186$ in the SAG model for all types.

We compare the real time auditor's expected utility for each triggered alert between the OSSP (the optimal objective value of LP (4.3)) and both the *offline* and *online SSE* (the optimal objective value of LP (4.2)). The offline SSE corresponds to the traditional method, which determines the auditing strategy at the end of the auditing cycle. By contrast, the online SSG determines the auditing strategy for each alert in real time, which is equivalent to an SAG without signaling.

One significant challenge in comparing the OSSP with the online SSE is that the real time budget consumption in the SAG is determined by the sampling result of warning/no

warning and, thus is not deterministic. This leads to a situation where, for the time series of alerts in each audit cycle, if there is no intervention, then the online SSG and the SAG will move independently with respect to the game status. As such, their performance cannot be directly compared. To set up a well-controlled environment for comparison, for each incoming alert we focus on the online SSG with its game status be the same as the current SAG instance. Recall that for the SAG, the auditor needs to reserve a portion of the total auditing budget for inspecting the repeated data requests at the end of each audit cycle. Due to the fact that it is unnecessary for the online SSG, we set the available budget at each incoming alert in the online SSG to be equal to the sum of the available budget of the SAG instance at the current time point, and the reserved budget of the SAG for the additional inspection of the repeated data requests. By doing so, it makes our comparison fair.

To investigate the robustness of the results over different game conditions, we evaluate the performance by varying three factors. First, we vary the loss value for the auditor with respect to each quit of a normal user when receiving a warning message. We set $C_t = \{-1, -5, -10\}$.[6] Second, to deter the attacker who quits until they receive no warning in the safe period for an SAG (where $p_0^{t*} = 0$ as shown in Theorem 5), we assess a series of constant budgets, which we set to $\alpha = \{1\%, 5\%\}$ of the total available budget B. We do not consider this situation in the baseline strategies because such loss does not apply. Third, we vary the total auditing budget. Specifically, we consider $B = \{30, 50, 70\}$. By setting $B = 50$, the available budgets for the SAG at the very beginning time point of an audit cycle are 49.5 for $\alpha = 1\%$ and 47.5 for $\alpha = 5\%$, respectively.

Considering the fact that the estimated payoff structure may not be perfect, we also test the robustness of the results by varying the values in the given payoff structure. To do so, we use $U_{a,c}$ and $U_{d,c}$ from the first type because these variables are more challenging for domain experts to articulate. We evaluate the performance by setting $U_{a,c}^1 = \{-500, -1000, -1500, -2000, -2500, -3000, -3500\}$ and fixing the other variables to

[6] To the best of our knowledge, there is no perfect measure for this loss in the EHR application domain.

their values in Table 5.2. We set $U^1_{d,c} = \{25, 50, 75, 100, 125, 150, 175\}$ and run the same evaluation as described above.

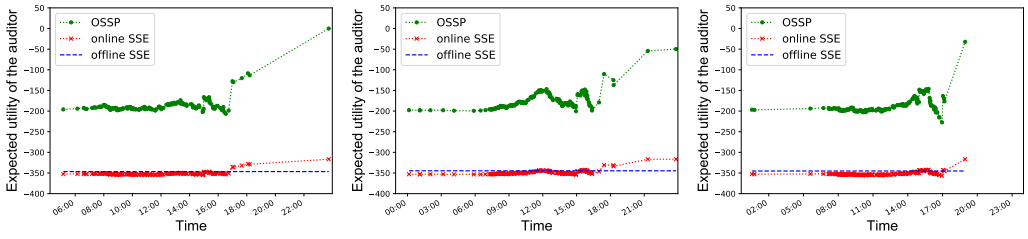
4.5.3 Results

We considered all 7 alert types described in Table 5.1. Due to space limitations, we only show the sequential results of 15 sequential testing days along the timeline in Figures 5.4a-4.5o by applying $B = 50, C_t = -1$ for all types and $\alpha = 1\%$.

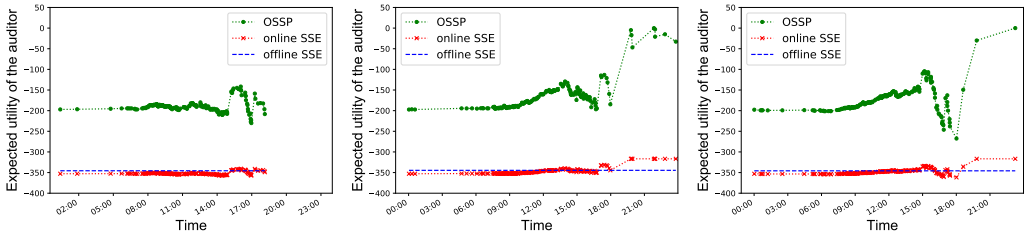
It is noteworthy that the type for each alert may not be aligned with the optimal attacking type in the OSSP strategy. Thus, to compare the approaches, we only apply the SAG on alerts whose type is equal to the best attacking type in the OSSP. For alerts whose types differ, we simply apply the online SSE strategy and use its optimal coverage probability to update the real time available budget. When applying SAGs, we first optimize the signaling scheme, then randomly sample whether to send a warning according to $\mathbf{P}(\xi_1^\tau)$. Next, we update (in real time) the available budget based on the signal.

Figures 5.4a-4.5o illustrate the real time expected utility of the auditor. It can be seen that the majority of alerts were triggered between $8:00AM$ and $5:00PM$, which generally corresponds to the normal working hours of VUMC. After this period, the rate of alerts slows down considerably. Note that the trend for offline SSE is flat because, in this method, the auditor's expected utility is the same for each alert regardless of when it is triggered.

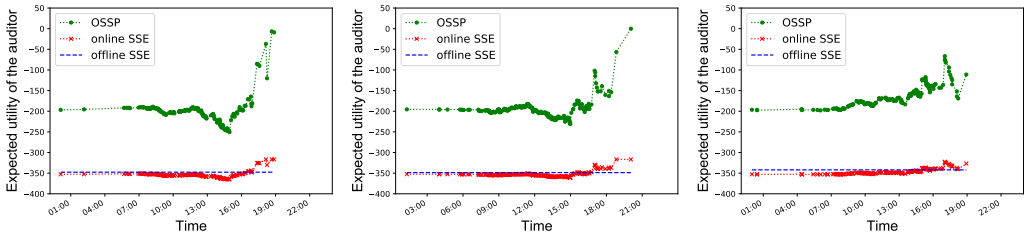
There are several notable findings and implications. First, in terms of the expected utility of the auditor, OSSP significantly outperforms the offline SSE and the online SSE. This suggests that the SAG increases auditing effectiveness. We believe that this advantage is due to the optimized signaling mechanism, which ensures the loss of the auditor is zero when sending warning messages. Second, at the end of each testing day, the auditor's expected utility for each approach does not drop below the online SSE. We believe that this is an artifact of the knowledge rollback, which slows down the budget consumption in this period. In particular, at the end of multiple testing days, such as illustrated in Figures



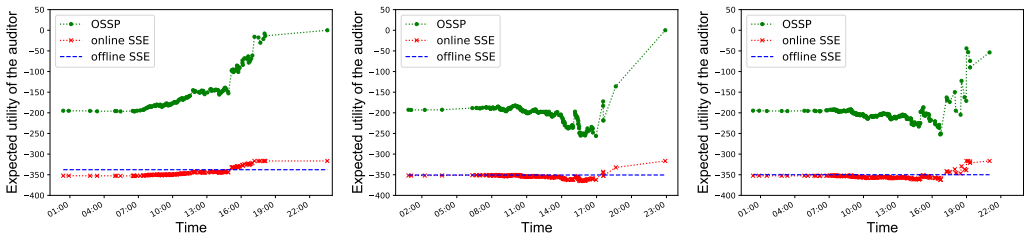
(a) Day 1 (b) Day 2 (c) Day 3



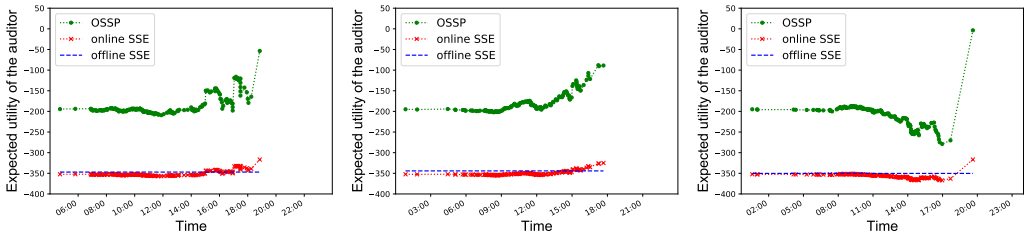
(d) Day 4 (e) Day 5 (f) Day 6



(g) Day 7 (h) Day 8 (i) Day 9



(j) Day 10 (k) Day 11 (l) Day 12



(m) Day 13 (n) Day 14 (o) Day 15

Figure 4.5: The auditor's expected utility in the OSSP and alternative equilibria for the 7 alert types with a total budget of $B = 50$. We applied $\alpha = 1\%$ and $C_t = -1$ for the OSSP.

Table 4.3: The advantages of OSSP over online SSE in terms of the mean (and the standard deviation) of the differences in the auditor's expected utility (15 testing days).

B	$C_t = -1$		$C_t = -5$		$C_t = -10$	
	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 5\%$
30	60.87 ± 28.31 15.99%	47.01 ± 32.17 12.45%	40.43 ± 23.95 10.59%	29.89 ± 28.77 7.92%	26.91 ± 25.77 7.06%	10.94 ± 24.93 2.90%
50	165.83 ± 24.49 47.26%	147.51 ± 27.74 42.65%	143.19 ± 33.98 40.87%	117.52 ± 34.56 34.20%	127.31 ± 37.55 36.23%	106.21 ± 38.85 31.21%
70	252.57 ± 20.44 77.31%	235.14 ± 23.57 72.87%	227.59 ± 33.10 69.31%	204.33 ± 36.77 63.63%	225.35 ± 37.58 68.73%	198.69 ± 40.93 61.89%

Table 4.4: The advantages of OSSP over online SSE in terms of the mean (and the standard deviation) of the differences in the auditor's expected utility. Asterisks indicate the original values we used in the evaluations above.

$\mathbf{U^1_{a,c}}$	-500	-1000	-1500	-2000^*	-2500	-3000	-3500
MEAN	67.89	120.28	148.29	167.64	180.67	184.99	194.54
STD	27.89	31.98	27.51	26.20	25.68	36.34	39.93
$\mathbf{U^1_{d,c}}$	25	50	75	100^*	125	150	175
MEAN	173.71	169.30	166.93	165.20	163.65	160.19	158.13
STD	26.33	25.32	25.58	24.92	23.93	24.61	23.36

5.4a, 4.5f, 4.5g, 4.5h, 4.5j, 4.5k and 4.5o, the expected auditor loss approaches 0. Third, the sequences of online SSE are close to the corresponding offline SSE sequences. This indicates that the auditing procedure does not benefit from determining only the coverage probability for each of the alert types in real time. In other words, the signaling mechanism in the SAG can assist the auditing tasks in various environments. Moreover, the advantage of OSSP over online SSE grows with the overall budget.

We expanded the investigation to consider various conditions of the auditing tasks. We computed the mean (and standard deviation[7] of) differences between the OSSP and the corresponding online SSE for each triggered alert across 15 testing days by varying the total auditing budget, the loss of the auditor on each quit of normal users, and the percentage of the budget for inspecting anomalous repeated requests. The results are shown in Table 4.3, where we also indicate the percentage of the averaged improvement in each setting. Here, this value is defined as the absolute improvement on the expected utility of the auditor divided by the optimal auditor's expected utility in the online SSE. From the results, we have the following significant observations. First, it is notable that OSSP consistently outperforms the online SSE with respect to the auditor's expected utility in a variety of auditing settings. For example, in the setting that $C_t = -1$ for all t and $\alpha = 1\%$, as B grows from 30 to 70, the auditor's expected utility improvement grows from 16% to

[7]Note that the distributions are not necessarily Gaussian. The standard deviations are largely dominated by the ending periods of testing days, where the expected utility of the auditor in the OSSP is usually close to 0.

77%. This is a trend that holds true for other settings as well. Second, by fixing B and C_t for all t, the auditor's expected utility decreases when we reserve more budget to investigate the repeated requests by single user. Yet, this is not unexpected because this approach reduces the amount of consumable auditing resources. Third, by increasing the cost of deterring a single normal data request, we also weaken the advantages of OSSP over the online SSE (when B and α are held constant).

We then investigated the robustness of the advantage of OSSP over online SSE by varying $U^1_{a,c}$ and $U^1_{d,c}$. We computed the mean (and standard deviation of) differences between the OSSP and the corresponding online SSE across all testing days. Here, we applied $B = 50, C_t = -1$ for all types and $\alpha = 1\%$. As can be seen in Table 4.4, OSSP maintains its advantage for a wide range of $U^1_{a,c}$ and $U^1_{d,c}$. The advantage of OSSP is inversely proportional with $U^1_{a,c}$ and directly proportional with $U^1_{d,c}$. Thus, even if the estimates of the payoff structure are imperfect, the SAG still outperforms baseline methods.

Table 4.5: The mean and standard deviation of auditor's expected utility at OSSP as a function of P^t (15 testing days).

C_t	P^t for all t		
	$\times 1.0$	$\times 0.5$	$\times 0.1$
-1	-185.54 ± 29.85	-179.92 ± 32.71	-175.47 ± 32.97
-5	-208.60 ± 41.91	-201.34 ± 33.92	-180.85 ± 32.75

Next, we considered how the probability of being scared away for normal users (i.e., P^t) influences the auditor's expected utility. Recall that, in the experiments reported on so far, we adopted $P^t = 0.186$, an estimate based on an environment that relied upon an unoptimized signaling procedure. However, this value can change in practice for several reasons. First, an optimized signaling scheme will likely influence users' access patterns, such as the frequency of triggering alerts, as well as how users respond to a signaling mechanism. Second, the probability P^t can decrease, if an organization effectively performs policy training with its employees, such that normal users may be less likely to be scared away if they receive a warning message when requesting access to a patient's record. Table 4.5 shows the

expected utility of the auditor at OSSP by varying the input of P^t in the setting of $B = 50$. We apply three values of P^t by reducing the original value to its $100\%, 50\%$ and 10%. It can be seen that the auditor's expected utility under OSSP improves as P^t reduces. When holding C_t constant, a t-test reveals that each pair of performances is statistically significantly different with $p < 10^{-6}$. This indicates that reducing the frequency of quitting for normal users reduces the usability costs and, thus, improves the auditing efficiency.

In addition, we tested the average running time for optimizing the SAG on a single alert across all the testing days. Using a laptop running Mac OS, an Intel i7 @ 3.1GHz, and 16GB of memory, we observed that the SAG could be solved in 0.06 seconds on average. As a consequence, it is unlikely that system users would unlikely perceive the extra processing time associated with optimizing the SAG in practice.

4.6 Discussion

In this chapter, we integrated signaling into auditing frameworks. We strategically warn the attacker in real time and then realize the audit strategy at the end of the audit cycle with an offline mode. In particular, we formalized the usability cost in our approach to model the real-world audit scenario. We further illustrated that such a defensive strategy improves the performance of defenders over existing game theoretic alternatives using real EHR auditing data. Our framework is generalizable to more powerful attackers because as long as the adversarial behavior can be represented by pattern(s), it will fit into our model. As such, our audit model is applicable to any capability of the attacker.

There are several limitations we wish to highlight as opportunities for future investigations. First, in this chapter we assumed that the attacker has a fixed payoff structure in each audit cycle; however, in practice, there may exist many types of attackers who can receive different utility on the same target. As a next step, we believe that the SAG can be extended for a Bayesian setting where the payoff structure of the attacker varies according to types. Second, in this chapter we focused on a single-attacker scenario. However, our model can

handle the multi-attacker scenario in which the attackers share the same payoff structure and act independently. In this case, the optimal strategy of the auditor for each alert is the same as in the single-attacker scenario. Moreover, it has been shown that solving problems that involve multiple types of attackers or collusion among them is NP-hard even if we do not consider signaling [90]. Third, in this chapter, we assumed that the attacker is perfectly rational. This is a strong assumption and may lead to an unexpected loss in practice. Thus, a more robust version of the SAG will be needed for wide deployment. Fourth, in this study we simplify modeling the dependence between alerts and assume that they are triggered independently. However, this may not be the case all the time in practice and attacks may evolve. Fifth, the scalability of solving the SAG, with respect to the number of alert types, needs more investigation in future.

4.7 Conclusion

Alert-based auditing is often deployed in database systems to address a variety of attacks to the data resources being stored and processed. However, the volume of alerts is often beyond the capability of administrators, thus limits the effectiveness of auditing. Our research illustrates that strategically incorporating signaling mechanisms into the data request workflow can significantly improve the auditing work. We investigated the features, as well as, the value of a game theoretic Signaling Audit Game, along with an Online Stackelberg Signaling Policy to solve the game. While we demonstrated the feasibility of this approach with the audit logs of an electronic medical record system at a large academic medical center, the approach is sufficiently generalized to support auditing in a wide range of environments. Though our investigation illustrates the merits of this approach, there are certain limitations that provide opportunities for extension and hardening of the framework for real world deployment.

Chapter 5

Robust Bayesian Signaling Games for Database Access Auditing

Inappropriate accesses to sensitive data stored and processed in database systems pose a threat to personal privacy and an organization's support of critical services. Alert-based auditing has been widely deployed to mitigate this threat. Recently, the problem of efficiently designing alert-based auditing schemes was modeled as a Signaling Audit Game (SAG), where users are warned of potential investigations into their activities. However, this model assumes that 1) attacker's goals are known to the defender, and 2) the attacker is a perfectly rational utility maximizer. Both assumptions, however, are likely violated in practice, and as a consequence, can cause an excessive loss to the defender (auditor). We introduce a new auditing framework, which we call a *robust Bayesian SAG*, which explicitly models the auditor's uncertainty about the attacker's goals and level of rationality. This new model integrates two types of robust modeling techniques: 1) bounding the worst-case deviation of an attacker's selected strategy from the optimal and 2) constraining the impact of the attacker's deviation on the auditor's loss. We then introduce several algorithmic approaches to compute robust solutions, the performance of which we evaluate in two environments: 1) the audit logs of over $10M$ real electronic health record accesses from a large academic medical center and 2) a simulated (controlled) environment derived from the real data. Further, we investigate the theoretical properties of these solutions and their relationship. We demonstrate in both environments that our robust solutions largely improve the performance of database access auditing compared to the state-of-the-art method. It is also notable that our solving algorithms take imperceptible running time to human and can scale for real time auditing.

5.1 Introduction

Personal data is increasingly recorded with finer granularity and greater quality. This makes data more useful for research and refinement of services, but also heightens privacy concerns [139, 121, 140, 141]. In particular, the sensitive nature of personal information can attract the attention of malicious attackers, who may benefit either directly (e.g., simply by gaining knowledge about someone else) by reusing the information to commit fraud (e.g., via identity theft) or indirectly by selling the information to others (e.g., the paparazzi). Even more of a concern is that, unlike attacks against the continuous operation of database management systems that can be quickly discovered, and subsequently fixed, attacks against data privacy are often initially silent, such that they may not be recognized until major losses have already occurred.

In recognition of this problem, a logging system with an alert functionality often operates in tandem with the primary system to detect and notify administrators about the potential data misuses incurred during daily use [125, 142]. In many systems, alerts are based on a set of rules that are predefined by administrators, each corresponding to a semantic type of a potentially malicious situation [25, 143]. These rules can be quite heterogenous, ranging from simple declarative statements (e.g., a user accessed a certain type of sensitive data) [144] to machine learning models (e.g., an automated outlier detection tool based on deep learning) [145]. Once alerts have been triggered, they are then brought to the attention of administrators for consideration in retrospective investigations. It is notable that, in many mission-critical systems, formally rigorous access control frameworks are often only weakly applied (if at all) to ensure the continuity of the workflows and operations. This makes retrospective auditing more important for mitigating the magnitude of a privacy breach. For instance, many healthcare organizations (HCOs) heavily rely on auditing procedures (instead of access control) to screen suspicious accesses to electronic health records (EHR) by their employees (and other privileged users) who may violate data use policies [129].

However, it is often the case that the amount of resources required for auditing (e.g., the time of security administrators or privacy officials) to investigate triggered alerts is substantially beyond what is available in practice [146]. Moreover, auditing is further hindered by the fact that many of the alerting systems that have been instituted are plagued by high false positive rates [147]. It adds much more complexity that attackers have a strong incentive to perform attacks strategically to mitigate the chance of being caught via observing and analyzing the established auditing mechanism. This can unfortunately make a strong traditional anomaly detection strategy fail.

To address such challenges, the Stackelberg security game (SSG) [34, 148, 86] has been adapted to model the interactions between a defender (auditor) and a set of attackers in a variant of the SSG, known as the *audit game* [98, 149, 36]. In an audit game, an attacker maximizes their expected utility based on the observation of the initial strategy committed to by an auditor. The auditor, taking the attacker's response space into consideration, then maximizes their expected utility by strategically assigning auditing resources to potential targets. Beyond the conventional model of auditing, we introduced a new concept in Chapter 4, *online signaling*, into the audit game, resulting in a *signaling audit game (SAG)*, which significantly outperforms previous models with respect to the auditor's expected utility [38]. Specifically, when an alert is triggered by a suspicious access request, the system can, in real time, send a warning to the requestor via, for example, a private pop-up window. At this point, the attacker has an opportunity to re-evaluate their expected utility and make a decision about whether or not to continue with an attack. The SAG optimizes both the warning and the audit decision simultaneously in real time.

Though it provides an excellent mechanism to deter attackers, there are two major deficiencies in the current structure of a SAG. First, they only consider one type of attacker in the system. In other words, they assume that all attackers share the same goal of attacks and, thus, the same payoff structure (i.e., rewards and penalties). This is unlikely to be the case in practice because different attackers often have different incentives and, thus,

utility functions. In the EHR misuse scenario, for instance, an attacker can be financially (extrinsically) motivated to sell a patient's information to a black market or the dark web, facilitating various forms of identity theft, such as fraudulent insurance claims [150]. Yet a different attacker might be intrinsically motivated to simply learn about the private medical condition of a high-profile patient or acquaintance. Importantly, attackers, when caught, will be penalized according to the severity of the committed breach. As such, the attacker types that an auditor may potentially face can largely influence the defense strategy and, thus, the expected utility of the auditor. An oversimplified model of attacker types may cause excessive loss of the auditor, leading to a failure of a SAG.

A second problem with the SAG is that it assumes that the attacker acts under perfect rationality, such that they always consider all possible strategies and choose the best one as their response. Though this has become a standard assumption in security game modeling, it rarely holds true for human attackers in the real world. And when human attackers are not utility maximizers, their decisions deviate from the optimal strategy [151, 152, 109, 153]. Failure to account for this fact in modeling may cause unexpected loss in the auditor's utility [39]. In other words, the selection of the auditor's defense strategy should be robust to attackers who are not perfectly rational.

In this paper, we address these deficiencies by introducing a new auditing framework, which we call a *robust Bayesian SAG*. This framework integrates two components. First, to account for multiple attacker types, as well as uncertainty in the auditor's belief against the actual type of attacker they face, we model the problem as a variant of a Bayesian Stackelberg game [154, 152], which we efficiently solve through a compact formulation. Second, to account for imperfect rationality of players, we integrate two types of methods to support robust optimization: 1) bounding the worst-case deviation of an attacker's strategy selection from their optimal strategy and 2) constraining the impact of the attacker's deviation on the auditor's loss. We incorporate each type of constraints into an algorithm for solving the robust Bayesian SAG in real time and create a corresponding solution con-

cept for each. We then investigate the theoretical properties of these solution concepts and the relationship between them.

We evaluate the performance of the robust Bayesian SAG in two environments: 1) a real environment associated with the audit logs of over 10 million real EHR accesses from Vanderbilt University Medical Center (VUMC) and 2) a simulated controlled environment derived from the real data. We specifically evaluate the expected utility of the auditor between the proposed solutions and the state-of-the-art auditing method in different conditions to demonstrate the value of the new auditing solutions and their scalability.

5.2 Preliminary and Notations

In this section, we review the online signaling mechanism in database access auditing (i.e., SAG) as well as its solution. Though we situate our presentation in the context of EHR misuse auditing, it should be recognized that the formulation can be applied to any database access auditing scenario. One of the reasons we focus on EHR auditing is that healthcare is the only industry where insiders are considered the greatest threat to the organization [155]. Moreover, this is a rapidly growing problem. In 2020, over 8.5 million health records were exposed in insider data misuse incidents in the US, a magnitude similar to all incidents reported between 2017 and 2019 [156].

In an HCO, a patient's data, including personal identifiable information and health related information, is stored and processed in a centralized system. The employees (or EHR users) provide healthcare services by accessing and updating the EHRs of patients. The interactions of a user and an EHR system consists of four fundamental steps:

- **STEP 1**: A user issues a search for a record. This search may be direct (using a patient's record number) or indirect (e.g., via a search for name and date of birth), in which case a set of patient records are served for further consideration.

- **STEP 2**: The user identifies the patient profile of interest and requests access to the

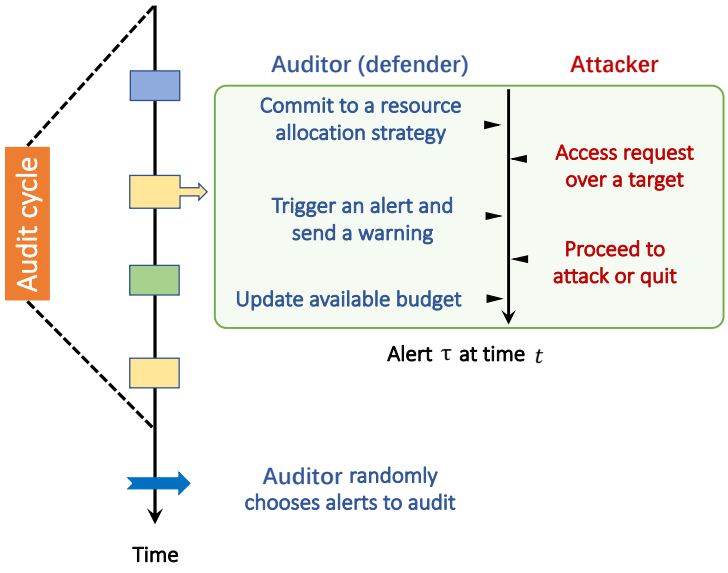
corresponding EHR.

- **STEP 3**: The system returns the requested record.
- **STEP 4**: The user interacts with the returned EHR.

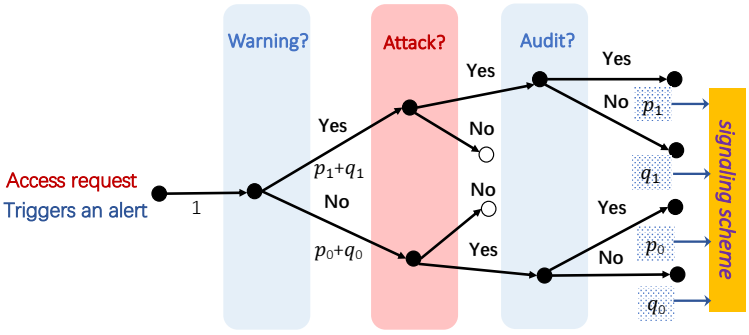
A SAG is played in real time between an *auditor* and an *attacker* within a predefined audit cycle, as shown in Fig. 5.1a. The auditor assumes each incoming alert is triggered by an attacker such that all interactions shown are carried out each time. For each access request that triggers an alert in real time by the misuse detection system, the auditor needs to determine: 1) which signal to send to the requestor in *real time* (e.g., warn the requestor or not), and 2) whether or not to audit the alert at the end of the audit cycle. The warning sent to the data requestor can take many forms, but it is typically presented as a message along the lines of "Your access might be investigated. Proceed or quit?", along with one button for *Proceed* and the other for *Quit*. The requestor can then click the button corresponding to their decision. When no warning is sent (or silent signal), the requested data will be returned to the requestor automatically without any further interaction. This process depends on four probabilities, as shown in Fig. 5.1b, which are defined as the *signaling scheme*.

Formally, p_1 denotes the joint probability that 1) a warning is sent to the requestor regarding the triggered alert, and 2) this alert will be investigated by the auditor. By contrast, q_1 is the joint probability that 1) a warning is sent to the requestor, and 2) this alert will not be investigated. Similarly, p_0 and q_0 are defined in the scenario where no warning is sent to the requestor. As a result, the probability of sending a warning is $p_1 + q_1$ and the probability that an alert will be investigated–regardless of warning or not–is $p_1 + p_0$. Due to the fact that there exist multiple predefined alert types, each of which corresponds to a potential type of violation (or attack type), the signaling scheme is designed to be alert type specific. We use $\{p_1^t, q_1^t, p_0^t, q_0^t\}$ to represent the signaling scheme of alert $t \in T$, where T is a finite set of alert types. Alerts with the same type are considered equivalent in terms of

the loss and reward to players.



(a) An illustration of the SAG over time. Each block in the timeline denotes a series of interactions for an alert triggered by the system. Different colors represent different alert types.



(b) The decision tree of players and the signaling scheme to be optimized in SAG. White nodes represent the end points that are not in the space of the signaling scheme.

Figure 5.1: Interactions between auditor and attacker in SAG.

We define the payoff structures (i.e., quantified utility in terms of rewards and penalties) of players by $\{U^t_{d,c}, U^t_{d,u}, U^t_{a,c}, U^t_{a,c}\}_{t\in T}$, where d and a indicate defender (auditor) and attacker, respectively, and c and u represent the scenarios where an attack is covered (or investigated) and not covered, respectively. If an alert of type t is indeed an attack, and it is not audited, then the auditor and the attacker will receive utility $U^t_{d,u}$ and $U^t_{a,u}$, respectively.

In the real world, it naturally holds true that $U^t_{a,c} < 0 < U^t_{a,u}$ and $U^t_{d,c} \geq 0 > U^t_{d,u}$.

The audit task for each audit cycle is constrained by an auditing budget B. We use B_τ to represent the available budget when alert τ of type t is triggered. Let δ^t be the probability of auditing alerts of type t and let d^t be the number of alerts associated with this type. Naturally, the budget constraint $\sum_t \delta^t V^t d^t \leq B_\tau$ will be satisfied, where V^t is the cost to audit an alert of type t. Note that the available budget is updated after each round of interactions (as shown in Figure 5.1a). Specifically, if a warning signal was sent to the requestor for alert τ, then the available budget for the next alert $\tau+1$ becomes $B_{\tau+1} = B_\tau - p^t_1/(p^t_1 + q^t_1) \cdot V^t$. By contrast, if there was no warning sent out for alert τ, then the budget becomes $B_{\tau+1} = B_\tau - p^t_0/(p^t_0 + q^t_0) \cdot V^t$.

To optimize the signaling schemes for each triggered alert and the budget allocation strategy over all alert types in an online manner, *Yan et al.* proposed a solution based on the budget constraint—*Online Stackelberg Signaling Policy (OSSP)* [38]. The core of OSSP is the following set of constraints:

$$\mathbb{E}^t_a(util|warning) = \frac{p^t_1}{p^t_1 + q^t_1} \cdot U^t_{a,c} + \frac{q^t_1}{p^t_1 + q^t_1} \cdot U^t_{a,u} \leq 0 \;\; \forall t \in T,$$

which forces the attacker's expected utility over each target to be non-positive. In other words, this setup ensures that the attacker's best response strategy to a warning is to quit. In this scenario, both players will receive zero utility. As such, the expected utility for the attacker and the auditor is:

$$\mathbb{E}^t_{a/d}(util) = p^t_0 \cdot U^t_{a/d,c} + q^t_0 \cdot U^t_{a/d,u}.$$

The OSSP is derived from the strong Stackelberg equilibrium [157, 86], which assumes that the attacker will break ties in favor of the defender. Thus, the OSSP is computed by solving multiple linear programs (LP), each assuming a distinct alert type is the best strategy for the attacker. The solution is the one that produces the largest expected utility

for the auditor.

To summarize, the SAG is essentially a leader-follower game that is a unique variant of the Stackelberg security game (SSG) [110]. The SAG leverages the time gap between moves of players and the potential impact of information exchange during this time, which induces a larger action space than a typical SSG and provide an opportunity to favor the auditor using their information advantage. Accordingly, the OSSP is a variant of a Strong Stackelberg equilibrium that is specific to a SAG, where 1) the auditor commits to a randomized joint signaling and auditing strategy in real time and 2) the attacker decides first about which alert type to induce and, subsequently, whether to proceed when receiving a warning.

Though a SAG provides mathematically effective auditing strategy, it is limited for practical use. This is because it oversimplifies the practical scenario where there could be more than one attacker types with distinct goals, each exhibiting a different payoff structure for the same target. In addition, the SAG neglects the fact that attackers often function under imperfect rationality. A failure to consider either of these facts in deriving audit solutions can lead to excessive loss for the auditor.

5.3 Robust Bayesian SAG

In this section, we formalize, and then build the solutions for, the robust Bayesian SAG in the general context of information services. We start with modeling a Bayesian version of SAG, which serves as a foundation for the robustness modeling later.

5.3.1 Bayesian SAG

We integrate the concept of *multiple player types* into a SAG [38] and refer to the resulting game as a *Bayesian SAG*. We then design its solving algorithm by combining the algorithm solving SAG [38] and the algorithm *DOBSS* [158] for solving Bayesian security games [87]. The Bayesian SAG still assumes the perfect rationality of players and both

players rigorously select their strategies that optimize their expected utilities. We assume that there is only one type of auditor and multiple types of attackers (i.e., all attackers share the same goal). We use θ^l to denote the priori probability that an attacker of type $l \in L$ appears in the system, where L is a finite set of attacker types. The pure strategy of an attacker (i.e., which alert type to use) of type l is denoted by binary variables z_t^l. The auditor's expected utility is $\sum_t \sum_l \theta^l z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$, where we add superscript l to the payoff notations to indicate the corresponding type of attacker.

It should be recognized that the auditor is subject to an additional loss that is caused by non-malicious users who walk away upon seeing a warning signal (i.e., choose "Quit" instead of "Proceed") when they properly request to access a sensitive data record that triggers alert(s). This usability cost is an important factor to consider in optimizing the signaling scheme. Following the usability cost design in [38], we update the auditor's expected utility for an arbitrary time point as $\mathbb{E}_d^\tau(util) = \sum_t \big(\sum_l \theta^l z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l}) + (p_1^t + q_1^t) \cdot P^t E_\tau^t C_t \big)$. Here, the usability cost, i.e., the second term, is proportional to 1) the probability that a warning is sent to the data requestor when an alert is triggered, 2) the probability that a non-malicious user walks away upon receiving a warning P^t, 3) the expected number of false positive alerts from the current moment in time (that is, when alert τ is triggered) to the end of the audit cycle E_τ^t, and 4) the loss (e.g., system efficiency) C_t due to each "walking-away" event from non-malicious users.

Let M be a very large positive number and A^l be the upper bound of the expected utility of an attacker in type l. The Bayesian SAG can be solved by applying the following mixed-integer quadratic program (MIQP):

$$\max_{p,q,B_\tau,z} \sum_t \left(\sum_l \theta^l z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l}) + (p_1^t + q_1^t) \cdot P^t E_\tau^t C_t \right)$$

s.t.

$$\begin{aligned}
&\forall t \in T, \forall l \in L, \quad && p_1^t U_{a,c}^{t,l} + q_1^t U_{a,u}^{t,l} \leq 0, \\
&\forall t \in T, \forall l \in L, \quad && 0 \leq A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \leq (1 - z_t^l) M, \\
&\forall t \in T, && p_1^t + p_0^t = \mathbb{E}_{d_\tau^t \sim D_\tau^t} \left(\frac{B_\tau^t}{V^t d_\tau^t} \right), \\
&\forall t \in T, && p_1^t + p_0^t + q_1^t + q_0^t = 1, \\
&\forall t \in T, && p_0^t, q_0^t, p_1^t, q_1^t \in [0,1], \\
&\forall t \in T, && B_\tau^t \in [0, B_\tau], \quad \sum_{t \in T} B_\tau^t \leq B_\tau, \\
&\forall t \in T, \forall l \in L, \quad && z_t^l \in \{0,1\}, \quad \sum_{t \in T} z_t^l = 1.
\end{aligned} \tag{5.1}$$

Instead of modeling in a fashion of multiple LPs, which is adopted by the algorithm proposed for solving SAG [38], we solve the Bayesian SAG using only one MIQP. This allows for a more compact game representation and, thus, more efficient search for an exact solution. The first constraint for the MIQP (5.1) forces the attacker to abort current access for all attacker types and targets. The left inequality of the second constraints ensures that A^l is the upper bound of the attacker's expected utility in type l. When an attacker of type l attacks target t as their response, then the right inequality is activated, which ensures that t is the best strategy for the attacker of type l. We follow the strategy in [38] to compute the probability that an attack of type t will be investigated by the auditor. In the third constraint, d_τ^t denotes the number of type-t alerts after the current alert τ is triggered til the end of the current audit cycle. The distribution of d_τ^t is assumed to follow a Poisson distribution D_τ^t, which can be easily learned from historical data. The last row of constraints limits the attacker strategy of any type to be a pure distribution (in comparison to be probabilistic) over all targets. Note that in the objective function we use the expected number of future alerts $\mathbb{E}_{d_\tau^t \sim D_\tau^t}(d_\tau^t)$ to approximate E_τ^t because the majority of alerts are false positives in the general audit setting. We refer to this solution concept a *Bayesian OSSP* because the

auditor only has uncertain knowledge about the attacker types they may encounter. We will use the Bayesian OSSP as a baseline in our experiments.

5.3.2 Imperfect rationality and robust strategies

In the real world, it is unreasonable to assume that attackers can always act with infallible utility maximizing rationality. It is also an unreasonably strong assumption that the attacker will break ties to favor the auditor in practice (which is unfortunately a premise of the OSSP and the Bayesian OSSP). This is because human attackers may not have the time, energy, or knowledge to perform accurate utility calculations to choose a strategy. In economics, such a phenomenon is defined as bounded rationality [159]. It has been empirically shown that a security game solution assuming perfect rationality of players may lead to unexpected loss in the face of human attackers because the defender underprotects those targets that they believe an attacker would not attack [152]. Now, we explore integrating two robust modeling methods, which serve as the basis of our solutions, to account for the imperfect rationality of attackers such that new solutions are robust to adversarial humans.

ε-robust: The first method is designed to explicitly bound the worst-case utility deviation of the attacker's strategy selection from their optimal strategy [152]. This can be achieved by modeling that attackers may select an ε-optimal strategy in the worst case, where $\varepsilon \geq 0$. In other words, attackers may respond with any strategy within ε deviation in utility from their optimal strategy. We use the following inequalities to formulate this method:

$$\varepsilon \cdot (1 - y_t^l) \leq A^l - (p_0^t \cdot U_{a,c}^{t,l} + q_0^t \cdot U_{a,u}^{t,l}) \leq \varepsilon + (1 - y_t^l) \cdot M,$$

where the binary variables y_t^l denote all ε-optimal strategies for attacker type l. Here, we allow an attacker to have more than one choice to attack. When $y_t^l = 1$ holds true, the inequalities above become $0 \leq A^l - (p_0^t \cdot U_{a,c}^{t,l} + q_0^t \cdot U_{a,u}^{t,l}) \leq \varepsilon$. This indicates that attacking target t is within the ε degradation from the corresponding utility of the best response. In

the other situation, where $y_t^l = 0$, the right inequality is deactivated, which implies that attacking target t causes a strategy to be more than ε away from the optimal solution and, thus, should be discarded.

β-robust: Instead of setting a clear boundary that an attacker can move, the second method is designed by constraining the maximum loss of the auditor caused by the deviation of the attacker's response from their optimal strategy [160]. Formally, it can be formulated as:

$$\gamma^l - (p_0^t \cdot U_{d,c}^{t,l} + q_0^t \cdot U_{d,u}^{t,l}) \leq \beta \cdot (A^l - (p_0^t \cdot U_{a,c}^{t,l} + q_0^t \cdot U_{a,u}^{t,l})),$$

where γ^l denotes the auditor's optimal expected utility without considering the usability cost in the face of an attacker in type l. The left side of this inequality denotes the auditor's loss when the attacker attacks target t (instead of choosing the best target). This value is a constraint factor of β times the attacker's loss for the deviation, where $\beta \geq 0$. This design does not set a hard cut-off point for attackers, but instead allows for a more gradual defense against an attacker's deviation.

We refer to these two special case games as the ε-robust Bayesian SAG and the β-robust Bayesian SAG.

5.3.3 Optimizing robust Bayesian SAG

We introduce two algorithms to solve the ε-robust Bayesian SAG and the β-robust Bayesian SAG as follows.

Solving the ε-robust Bayesian SAG. By applying an ε-optimal constraint to the attacker's expected utility, we solve the ε-robust Bayesian SAG through the following mixed-integer linear program (MILP):

$$\max_{p,q,B_\tau,z,y} \sum_{l\in L} \ \theta^l \gamma^l + \sum_{t\in T}(p_1^t + q_1^t)\cdot P^t E_\tau^t C_t$$

s.t.

$$\begin{aligned}
&\forall t \in T, \forall l \in L, \quad p_1^t U_{a,c}^{t,l} + q_1^t U_{a,u}^{t,l} \le 0, \\
&\forall t \in T, \forall l \in L, \quad 0 \le A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \le (1 - z_t^l)M, \\
&\forall t \in T, \forall l \in L, \quad \varepsilon(1 - y_t^l) \le A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \le \varepsilon + (1 - y_t^l)M, \\
&\forall t \in T, \forall l \in L, \quad (1 - y_t^l)M + p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l} \ge \gamma^l, \\
&\forall t \in T, \qquad\qquad p_1^t + p_0^t = \mathbb{E}_{d_\tau^t \sim D_\tau^t}\left(\frac{B_\tau^t}{V^t d_\tau^t}\right), \\
&\forall t \in T, \qquad\qquad p_1^t + p_0^t + q_1^t + q_0^t = 1, \\
&\forall t \in T, \qquad\qquad p_0^t, q_0^t, p_1^t, q_1^t \in [0,1], \\
&\forall t \in T, \qquad\qquad B_\tau^t \in [0, B_\tau], \quad \sum_{t\in T} B_\tau^t \le B_\tau, \\
&\forall t \in T, \forall l \in L, \quad z_t^l \le y_t^l, \quad z_t^l, y_t^l \in \{0,1\}, \\
&\qquad\qquad \forall l \in L, \quad \sum_{t\in T} z_t^l = 1, \quad \sum_{t\in T} y_t^l \ge 1.
\end{aligned} \tag{5.2}$$

Similar to the Bayesian SAG, the first constraint above ensures that when an attacker receives a warning message, it is in the best interest of the attacker to quit accessing the requested record to avoid getting a non-positive reward. Given attacker type l, the binary variables z_t^l denote the attacker's optimal strategy with an expected utility of A^l, and binary variables y_t^l denote all ε-optimal strategies for attacker type l. The second and third constraints, as well as constraints in last two rows, are used to fulfill these definitions. To achieve the robust auditing against an attacker's deviation, we maximize the lower bound of the auditor's expected utility, i.e., γ^l, for all possible deviations by the attacker to guard against the worst case scenario. The objective function also considers the usability cost for deterring the non-malicious users of the system. We refer to this solution concept as the *ε-robust Bayesian OSSP*.

Solving the β-robust Bayesian SAG. To directly constrain the maximum loss of the auditor in the face of imperfectly rational attackers, we introduce the following MILP algorithm:

$$\max_{p,q,B_\tau,z} \sum_{l\in L} \theta^l \gamma^l + \sum_{t\in T}(p_1^t + q_1^t)\cdot P^t E_\tau^t C_t$$

$$s.t.$$

$$\begin{aligned}
&\forall t\in T, \forall l\in L, \quad p_1^t U_{a,c}^{t,l} + q_1^t U_{a,u}^{t,l} \le 0,\\
&\forall t\in T, \forall l\in L, \quad 0 \le A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \le (1-z_t^l)M,\\
&\forall t\in T, \forall l\in L, \quad (1-z_t^l)M + p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l} \ge \gamma^l,\\
&\forall t\in T, \forall l\in L, \quad \gamma^l - (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})\\
&\qquad\qquad\qquad\qquad \le \beta(A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l})),\\
&\forall t\in T, \qquad p_1^t + p_0^t = \mathbb{E}_{d_\tau^t\sim D_\tau^t}\left(\frac{B_\tau^t}{V^t d_\tau^t}\right),\\
&\forall t\in T, \qquad p_1^t + p_0^t + q_1^t + q_0^t = 1,\\
&\forall t\in T, \qquad p_0^t, q_0^t, p_1^t, q_1^t \in [0,1],\\
&\forall t\in T, \qquad B_\tau^t \in [0, B_\tau], \quad \sum_{t\in T} B_\tau^t \le B_\tau,\\
&\forall t\in T, \forall l\in L, \quad z_t^l \in \{0,1\}, \quad \sum_{t\in T} z_t^l = 1.
\end{aligned} \tag{5.3}$$

Here, γ^l is used to represent the utility of the auditor (without the usability cost) against the type-l attacker's best strategy. In this regard, we neither explicitly model nor constrain the attacker's deviation from their optimal response strategy. Instead, we constrain the potential loss of the auditor as a function of the attacker's loss through the fourth constraint. We refer to this solution concept as the β-*robust Bayesian OSSP*.

Both algorithms for robust signaling schemes trade off the optimal utility of the auditor for protecting against the weakness of the Bayesian SAG in the face of real world attackers. As a result, there are similarities worth highlighting. Intuitively, a larger ε and a smaller β in a reasonable range can render more robustness to the auditing task. This implies

that both ε and β need to be properly tuned in the real auditing scenario according to its characteristics, and vice versa. However, these two methods have distinct perspectives to achieve robustness, which may demonstrate differences in their theoretical properties, practical protection efficiency, and scalability.

5.4 Theoretical Properties

In this section, we analyze the theoretical properties of the Bayesian OSSP, the ε-robust Bayesian OSSP, and the β-robust Bayesian OSSP solutions, as well as their relationship. They will be used as theory basis to explain experimental results. We start with a notable property of the Bayesian OSSP that the associated signaling scheme only sends warning for the best alert types (targets) from all potential attacker types.

Theorem 8 *For any Bayesian OSSP, if the set of alert types* $\mathsf{Q} = \{t_*^1, t_*^2, ..., t_*^K\}$*, where* $K < |T|$ *and* $\mathsf{Q} \subseteq T$*, consists of the best response strategies for all possible attacker types* ($\forall l \in L$)*, then* $p_1^t = q_1^t = 0$ *for* $\forall t \notin \mathsf{Q}$.

Proof Assume $\mathsf{S} = (\{p_0^t, p_1^t, q_0^t, q_1^t, \{z_t^l\}_{l\in L}, B_\tau^t\}_{t\in T}, \{A^l\}_{l\in L})$ is any optimal solution of MIQP (5.1). We define a set of new variables by letting $\bar{p}_0^{t_*} = p_0^{t_*}, \bar{p}_1^{t_*} = p_1^{t_*}, \bar{q}_0^{t_*} = q_0^{t_*}, \bar{q}_0^{t_*} = q_0^{t_*}$ for all $t_* \in \mathsf{Q}$, which are the same as in S; however, we define $\bar{p}_0^t = p_0^t + p_1^t, \bar{q}_0^t = q_0^t + q_1^t$ and $\bar{p}_1^t = 0, \bar{q}_1^t = 0$ for any $t \notin \mathsf{Q}$. In addition, we define $\bar{z}_t^l = z_t^l$ for all $t \in T, l \in L$, $\bar{A}^l = A^l$ for all $l \in L$, and $\bar{B}_\tau^t = B_\tau^t$ for all $t \in T$. We now prove that these newly defined variables will never reduce the objective value of MIQP (5.1) on its optimal solution S.

We first demonstrate the feasibility of the newly defined variables, i.e.,

$$\bar{\mathsf{S}} = (\{\bar{p}_0^t, \bar{p}_1^t, \bar{q}_0^t, \bar{q}_1^t, \{\bar{z}_t^l\}_{l\in L}, \bar{B}_\tau^t\}_{t\in T}, \{\bar{A}^l\}_{l\in L}).$$

The first constraint of MIQP (5.1) is still satisfied both for all $t \in \mathsf{Q}$ and for all $t \notin \mathsf{Q}$ since $\bar{p}_1^t = \bar{q}_1^t = 0$. As $\bar{z}_t^l$ remains the same for all $t \in T, l \in L$, the second constraint holds true for

any $t \in \mathsf{Q}$. For any $t \notin \mathsf{Q}$ (which corresponds to $\bar{z}_t^l = 0$ for all $l \in L$), the left inequality still holds true, because

$$\begin{aligned}
\bar{A}^l &- (\bar{p}_0^t U_{a,c}^{t,l} + \bar{q}_0^t U_{a,u}^{t,l}) \\
&= A^l - (p_0^t + p_1^t) \cdot U_{a,c}^{t,l} - (q_0^t + q_1^t) \cdot U_{a,u}^{t,l} \\
&= A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) - (p_1^t U_{a,c}^{t,l} + q_1^t U_{a,u}^{t,l}) \\
&\geq A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \;\geq\; 0,
\end{aligned}$$

where the first inequality above is due to $p_1^t U_{a,c}^t + q_1^t U_{a,u}^t \leq 0$ as a constraint of MIQP (5.1). Thus, the second constraint in MIQP (5.1) is feasible. It is evident that the rest of the constraints can be satisfied for the newly defined variables.

Next, we show that the newly defined variables do not harm the objective value that corresponds to the optimal solution S. The first term of the objective function can be rewritten as $\sum_t \sum_l \theta^l \bar{z}_t^l \cdot (\bar{p}_0^t U_{d,c}^{t,l} + \bar{q}_0^t U_{d,u}^{t,l}) = \sum_{t \in Q} \sum_l \theta^l \bar{z}_t^l \cdot (\bar{p}_0^t U_{d,c}^{t,l} + \bar{q}_0^t U_{d,u}^{t,l}) + \sum_{t \notin Q} \sum_l \theta^l \bar{z}_t^l \cdot (\bar{p}_0^t U_{d,c}^{t,l} + \bar{q}_0^t U_{d,u}^{t,l}) = \sum_{t \in Q} \sum_l \theta^l z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l}) = \sum_t \sum_l \theta^l z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$, which is the same as in S. This is because $\bar{z}_t^l = z_t^l = 0$ for any $t \notin Q$. The second term of the objective function can be transformed to $\sum_{t \in \mathsf{Q}} (\bar{p}_1^t + \bar{q}_1^t) \cdot P^t E_\tau^t C_t + \sum_{t \notin \mathsf{Q}} (\bar{p}_1^t + \bar{q}_1^t) \cdot P^t E_\tau^t C_t = \sum_{t \in \mathsf{Q}} (p_1^t + q_1^t) \cdot P^t E_\tau^t C_t$, which does not reduce the objective value, because $P^t E_\tau^t C_t \leq 0$. In summary, the newly defined variables yield a solution $\bar{\mathsf{S}}$ with a objective value that is no smaller than the original value with S. This proves the theorem. ■

Theorem 8 implies that if all types of perfectly rational attackers avoid targeting certain alert type(s), then the best strategy for the auditor is to turn off the signaling procedure for those types to prevent loss incurred by sending warnings. Theorem 8 cannot generally hold true for either the ε-robust Bayesian OSSP and the β-robust Bayesian OSSP. Intuitively, this is because these two robust game models shift their optimization power to account for possible deviations of attackers with imperfect rationality, where the auditor will lose less

than using the Bayesian OSSP. This is done by changing the signaling scheme. And, if the signaling procedures were still turned off for all suboptimal attack strategies, then the auditor can never benefit from the robustness design.

Next, in the following two theorems, we show that the ε-robust Bayesian OSSP and the β-robust Bayesian OSSP are equivalent in certain conditions.

Theorem 9 *Let* W *be the greatest absolute expected utility of an arbitrary attacker in* ε*-robust Bayesian SAG, if* $\varepsilon > 2\mathsf{W}$ *and* $\beta = 0$*, then the* ε*-robust Bayesian OSSP and the* β*-robust Bayesian OSSP are equivalent.*

Proof Note that A^l is the expected attacker's utility under the optimal response (i.e., $z_t^l = 1$), thus $|A^l| \leq \mathsf{W}$ for all $l \in L$ holds true. Given $\varepsilon > 2\mathsf{W}$ in ε-robust SAG, we first prove that $y_t^l = 1$ for all $t \in T, l \in L$ in any ε-robust Bayesian OSSP. To do so, we assume $\exists y_t^l = 0$, then the right inequality of the third constraint in MILP (5.2) turns inactive, whereas the left inequality becomes $p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l} \leq A^l - \varepsilon$. Then we have $A^l - \varepsilon \leq \mathsf{W} - \varepsilon < \mathsf{W} - 2\mathsf{W} = -\mathsf{W}$. According to the definition of W, $-\mathsf{W} \leq p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}$ for all $t \in T$. Then we have $-\mathsf{W} \leq p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l} < -\mathsf{W}$, which is a contradiction.

With $y_t^l = 1$ for all $t \in T, l \in L$, the third constraint of MILP (5.2) transforms to $0 \leq A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \leq \varepsilon$, where the left inequality is then equivalent to the left inequality of the second constraint. We now prove that $A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \leq \varepsilon$ for all $t \in T, l \in L$ by supposing $\exists t \in T, l \in L$, that $A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) > \varepsilon$ holds true. This can be rewritten as $A^l > (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) + \varepsilon > (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) + 2\mathsf{W} > -\mathsf{W} + 2\mathsf{W} = \mathsf{W}$. This contradicts to $|A^l| \leq \mathsf{W}$ for all $l \in L$. Thus, the third constraint of MILP (5.2) is trivially satisfied, which thus can be removed.

When $\beta = 0$, the fourth constraint of MILP (5.3) then becomes $\gamma^l \leq (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$ for all $t \in T, l \in L$, which implies the third constraint in MILP (5.3) (which thus can be removed), and is the same as the fourth constraint of MILP (5.2) given $y_t^l = 1$ for all $t \in T, l \in L$. As a consequence, MILP (5.3) shares the same constraints with MILP (5.2) in the

given two conditions. In addition, the objective functions of MILP (5.3) and (5.2) are the same. This theorem is proved. ■

Intuitively, this means that if ε is sufficiently large, then the ε-robust Bayesian SAG will cover all possible deviations of an attacker. This makes γ^l in MILP (5.2) the lower bound of the expected utility of an attacker in type l, which is indicated by the fourth constraint of MILP (5.2). The objective function is then translated to maximize the sum of the specified lower bound of the auditor's utility and the potential usability cost. We call this a *pseudo-MAXIMIN* solution. Similarly, if $\beta = 0$, the β-robust Bayesian SAG would not set constraints based on any specific deviation of the attacker, but instead makes γ^l the lower bound of the auditor's (in type l) expected utility (without usability cost). Thus, the β-robust Bayesian OSSP in this scenario becomes a *pseudo-MAXIMIN* solution as well. The pseudo-MAXIMIN solution seeks to achieve the highest robustness of auditing such that the auditor's expected utility in the face of a totally random attacker will not be unexpectedly low.

Theorem 10 *If $\varepsilon = 0$ and β is sufficiently large, then the ε-robust Bayesian OSSP and the β-robust Bayesian OSSP are equivalent and both degrade to the Bayesian OSSP.*

Proof We first show that the ε-robust Bayesian OSSP is equivalent to the Bayesian OSSP when $\varepsilon = 0$. The third constraint of MILP (5.2) becomes $0 \leq A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \leq (1 - y_t^l)M$ with $\varepsilon = 0$, which shares the same format with the second constraint. We then prove that $y_t^l = z_t^l$ for all $t \in T, l \in L$. Given $z_t^l \leq y_t^l$ and $z_t^l, y_t^l \in \{0,1\}$ for all $t \in T, l \in L$, we assume that $\exists t', l^\#$ such that $y_{t'}^{l^\#} = 1, z_{t'}^{l^\#} = 0$, which implies that $\exists t^* \neq t'$ such that $z_{t^*}^{l^\#} = 1$ (given $\sum_{t \in T} z_t^l = 1$) and $y_{t^*}^{l^\#} = 1$. Then for the attacker of type $l^\# \in L$, we have $0 \leq A^{l^\#} - (p_0^{t'} U_{a,c}^{t',l^\#} + q_0^{t'} U_{a,u}^{t',l^\#}) \leq (1 - y_{t'}^{l^\#})M = 0$. Then we have $A^{l^\#} = (p_0^{t'} U_{a,c}^{t',l^\#} + q_0^{t'} U_{a,u}^{t',l^\#})$. Given that $0 \leq A^{l^\#} - (p_0^{t^*} U_{a,c}^{t^*,l^\#} + q_0^{t^*} U_{a,u}^{t^*,l^\#}) \leq (1 - z_{t^*}^{l^\#})M = 0$, it is obvious that the attacker of type $l^\#$ is indifferent to attack a target of type t' and t^*, as each leads to the greatest expected utility, $A^{l^\#}$. Applying $y_{t^*}^{l^\#} = 1, y_{t'}^{l^\#} = 1$ to the fourth constraint of MILP (5.2), we get $p_0^{t^*} U_{d,c}^{t^*,l^\#} + q_0^{t^*} U_{d,u}^{t^*,l^\#} \geq \gamma^{l^\#}$ and $p_0^{t'} U_{d,c}^{t',l^\#} + q_0^{t'} U_{d,u}^{t',l^\#} \geq \gamma^{l^\#}$, implying that $\gamma^{l^\#}$ is a lower bound

of the auditor's expected utility in face of the attacker of type $l^{\#}$ with their best response t^* and t'. This would only serve to reduce the auditor's expected utility and thus would not be an optimal solution. As a result, $y_t^l = z_t^l$ for all $t \in T, l \in L$ holds true.

To show that the optimal objective value of MILP (5.2) is no less than that of of MIQP (5.1), we assume $(\{p_0^t, p_1^t, q_0^t, q_1^t, \{z_t^l\}_{l\in L}, B_\tau^t\}_{t\in T}, \{A^l\}_{l\in L})$ is any optimal solution of MIQP (5.1). We then define $\bar{p}_1^t = p_1^t, \bar{q}_1^t = q_1^t, \bar{p}_0^t = p_0^t, \bar{q}_0^t = q_0^t, \bar{z}_t^l = \bar{y}_t^l = z_t^l, \bar{A}^l = A^l, \bar{B}_\tau^t = B_\tau^t, \bar{\gamma}^l = \sum_{t\in T} z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$ for all $t \in T, l \in L$. All constraints in MILP (5.2) except the fourth constraint can be easily verified to hold true when applying the newly defined variables. The fourth set of constraints can be rewritten as $\forall t \in T, l \in L, (1 - z_t^l)M + p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l} \geq \sum_{t\in T} z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$. Given any $l \in L$, where $z_{t^*}^l = 1$ and thus $z_t^l = 0$ for all $t \neq t^*$, this inequality still holds true. Thus, the newly defined variables are a feasible solution of MILP (5.2). By applying $\bar{\gamma}^l$ to the objective function of MILP (5.2), which yields the same objective function with MILP (5.2), we can therefore obtain a same objective value using the new variables in MILP (5.2).

We then show that the optimal objective value of MIQP (5.1) is no less than that of MILP (5.2). To do so, we assume $(\{p_0^t, p_1^t, q_0^t, q_1^t, \{z_t^l = y_t^l\}_{l\in L}, B_\tau^t\}_{t\in T}, \{A^l\}_{l\in L})$ is any optimal solution of MILP (5.2). We define $\bar{p}_1^t = p_1^t, \bar{q}_1^t = q_1^t, \bar{p}_0^t = p_0^t, \bar{q}_0^t = q_0^t, \bar{z}_t^l = \bar{y}_t^l = z_t^l, \bar{A}^l = A^l, \bar{B}_\tau^t = B_\tau^t$ for all $t \in T, l \in L$. All of the constraints of MIQP (5.1) are satisfied. In the objective function of MIQP (5.1), the term $\sum_{t\in T} \bar{z}_t^l \cdot (\bar{p}_0^t U_{d,c}^{t,l} + \bar{q}_0^t U_{d,u}^{t,l}) = \sum_{t\in T} z_t^l \cdot (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l}) = p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l}$ given $z_t^l = 1$ for all $l \in L$, which is no less than γ^l because according to the fourth constraint of MILP (5.2), $z_t^l = y_t^l = 1$ makes this term greater or equal to γ^l. Thus, the ε-robust Bayesian OSSP is equivalent to the Bayesian OSSP when $\varepsilon = 0$.

Next, we show that the β-robust Bayesian OSSP is equivalent to the Bayesian OSSP if β is sufficiently large. In this case, the fourth constraint of MILP (5.3) is trivially satisfied, thus can be removed. We can then apply the same proof method above to show that a set of newly defined feasible variables for MIQP (5.1) based on any optimal solution of MILP

(5.3) can yield an objective value that is no less than the corresponding objective value of the β-robust Bayesian OSSP, and vice versa. ■

In other words, $\varepsilon = 0$ and a sufficiently large β both lead to the loss of robustness of the signaling audit system against any possible deviation of a potential attacker from their best response strategy. By contrast, as an implication of Theorem 9, the robustness level of the signaling audit system is pushed to the other end, where the worst case attacker (irrational at all) is considered through maximizing the lower bound expected utility of the auditor. In Theorem 11, we show how the values of ε and β between these two ends influence the optimal objective values of MILP (5.2) and (5.3).

Theorem 11 *Consider the non-negative ε and β as independent variables, the optimal objective value of MILP* (5.2) *is a monotonically non-increasing function of ε, whereas the optimal value of MILP* (5.3) *is monotonically non-decreasing function of β.*

Proof For MILP (5.2), the third constraint indicates that if an attacker of type l deviates from the optimal response strategy (which leads to a gain of A^l for the attacker) within ε regarding the expected utility (i.e., $A^l - (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l}) \leq \varepsilon$), then this attack strategy t can be represented by $y_t^l = 1$. It follows that for any $l \in L$, a larger ε ensures that the number of the attacker's strategies falling into ε degradation does not decrease at least (sometimes it can increase). Note that the number of active constraints in the fourth inequality is equal to the number of the attacker's strategies that are within ε degradation. This implies that the number of newly added active constraints to MILP (5.2) due to the increase of ε will not decrease at least (sometimes increase), which thus will at least not increase (sometimes decrease when there are more active constraints from the fourth constraint) the optimal objective value of MILP (5.2). As a result, the optimal objective value of MILP (5.2) is a monotonically non-increasing function of ε.

The fourth constraint of MILP (5.3) bounds the loss of the auditor due to the deviation of the attacker of type l from the optimal strategy (i.e., $\gamma^l - (p_0^t U_{d,c}^{t,l} + q_0^t U_{d,u}^{t,l})$) by the loss of the attacker because of the deviation with a coefficient of β. Thus, a smaller β leads to

a tighter set of constraints, which thus will at least not increase the optimal objective value of MILP (5.3) given that other constraints keep the same. This proves the second half of the theorem. ■

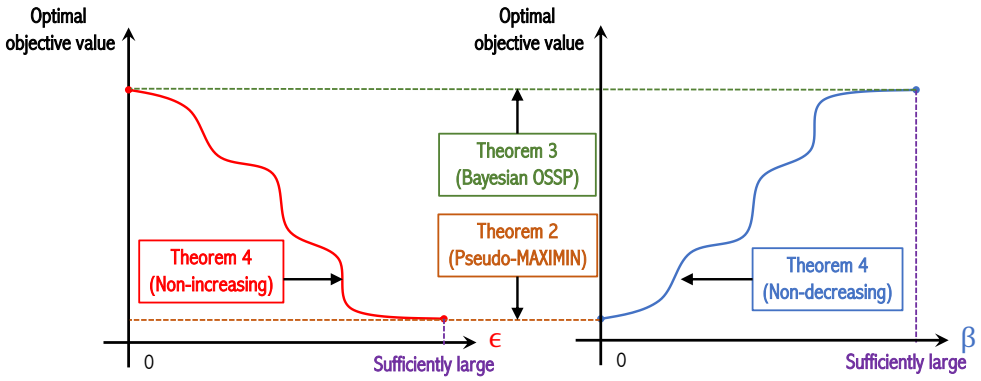


Figure 5.2: A graphical summary of Theorems 9, 10, 11. Note that the two curves in their own horizontal spaces are monotonically non-increasing and non-decreasing, respectively.

An important implication from Theorems 10 and 11 is that the optimal objective values of MILPs (5.2) and (5.3), which respectively correspond to the ε- and β-robust Bayesian OSSP, are no greater than the optimal value of MIQPs (5.1), which correspond to Bayesian OSSP, regardless of the non-negative ε and β. On the other hand, Theorems 9 and 11 imply that the optimal objective values of MILPs (5.2) and (5.3) can be no less than the optimal value when ε is sufficiently large and $\beta = 0$. Fig. 5.2 provides a graphical summary of the last three theorems. The two end points linked by the green dashed line indicate the largest objective value (the auditor's expected utility with the usability cost) and the two auditing solutions demonstrate no robust features at these points; however, the points linked by the brown dashed line indicate the smallest auditor's expected utility with the usability cost such that both auditing solutions consider the worst-case rationality of an attacker (i.e., a totally random attacker in selecting their strategy). It is evident that accounting for robustness sacrifices the auditor's optimal expected utility in the face of uncertain types of the attacker.

Table 5.1: A summary of alert types and their daily statistics.

ID	Description	Mean	St.dev
1	Same Last Name	196.57	17.30
2	Department Co-worker	29.02	5.56
3	Neighbor ($\leq$ 0.5 miles)	140.46	23.23
4	Same Address	10.84	3.73
5	Last Name & Neighbor ($\leq$ 0.5 miles)	25.43	4.51
6	Last Name & Same Address	15.14	4.10
7	Last Name & Same Address & Neighbor ($\leq$ 0.5 miles)	43.27	6.45

5.5 Model Evaluation

In this section, we evaluate the performance of the ε- and β-robust Bayesian SAG. We compare them with a non-robust baseline model (i.e., the Bayesian SAG, the adapted version of the solution in [38], which is the only state-of-the-art model to consider) in terms of the auditor's utility. The experiments are designed and conducted in two environments. The first is a reproduced real-time auditing environment using real EHR access logs from Vanderbilt University Medical Center (VUMC). By contrast, the second is a simulated controlled environment built on a real auditing scenario. This allows us to simulate attacker behaviors regarding the rationality degree. We test different auditing conditions by varying multiple key parameters. We begin with an introduction of our dataset and then describe the experimental setup for the two environments.

5.5.1 Dataset

The experiments in this study are based on a dataset of $10M$ real EHR access logs collected from the EHR system deployed at VUMC, which were also used in several earlier studies [38, 36]. These correspond to all EHR access events during a period of 56 continuous normal working days in 2017. Data for holidays and weekends were excluded because their access patterns are different from working days. The average number of daily unique access events (i.e., user A accesses patient B's EHR in one day) is $192K$ with a standard

Table 5.2: The payoff structures of the auditor (top) and the attacker (bottom) for the predefined alert types.

Payoff	**Attack (target) type**						
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$
$U_{d,c}^{t,1}$	100	150	150	300	400	600	700
$U_{d,c}^{t,2}$	10	15	20	30	35	50	70
$U_{d,u}^{t,1}$	-400	-500	-600	-800	-1000	-1500	-2000
$U_{d,u}^{t,2}$	-40	-45	-50	-60	-70	-85	-200
$U_{a,c}^{t,1}$	-2000	-2250	-2500	-2500	-3000	-5000	-6000
$U_{a,c}^{t,2}$	-100	-120	-180	-200	-250	-280	-330
$U_{a,u}^{t,1}$	400	400	450	600	650	700	800
$U_{a,u}^{t,2}$	10	10	15	20	30	40	75

deviation of $8.97K$. The information associated with each EHR access event includes the user's and patient's names, their residential addresses, and if a patient is also an employee of VUMC, and their department affiliations. All of the evens are timestamped. Four predefined alert types are used to trigger real time alerts which are recorded in our dataset: user and patient (whose data was requested) 1) share the same last name, 2) are co-workers from the same department, 3) are neighbors (≤ 0.5 miles), and 4) share the same residential address. If an EHR access event triggers more than one alert type, their combination is defined as a new alert type. The description of alert types used in this study and their daily statistics are provided in Table 5.1.

In this study, we focus on two types of attackers seen in practice, each with a distinct motivation for EHR misuse. The first type of attacker is financially motivated (referred to as *F-MOT*), which often leads to medical identity theft (such as insurance fraud) [161], while the second is motivated by intrinsic curiosity (referred to as *C-MOT*) about someone's clinical condition [162], including, but not limited to, medical conditions, treatment and medical visit history. The priori distribution of these two types that the auditor can encounter is set to $\theta^1 : \theta^2 = 2 : 8$. The payoff structures for both players under the two attacker types are shown in Table 5.2. We set the payoffs for the first attacker type to be

one order of magnitude higher than the second type because financially motivated attacks can lead to larger penalty and reward. These values are estimates based on discussions with experts working in the area. Note that though our dataset does not include straightforward rules for F-MOT attackers, this will not influence model evaluation.

5.5.2 Experimental Setup

1). Real environment

We follow the same scenario in [38] to define the audit cycle as one day from *0:00:00* to *23:59:59*. We apply a sliding window with a length of 42 days to the data collection period (i.e., 56 continuous working days) to construct 15 data groups. Each consists of a 41-day period of history observations for estimating the number of future alerts, along with the subsequent day for model evaluation purpose. For each evaluation day, a real time auditing environment is established where the auditor and EHR users interact with each other on the timeline. Instead of evaluating the objective values of all candidate game solutions, we compute the expected utility of the auditor conditioned on that a user of type $\tilde{l}$ (which is randomly assigned based on $\theta^1 : \theta^2$) requests an EHR which triggers an alert of type $\tilde{t}$ as follows:

$$p_0^{\tilde{t}} U_{d,c}^{\tilde{t},\tilde{l}} + q_0^{\tilde{t}} U_{d,u}^{\tilde{t},\tilde{l}} + \sum_t (p_1^t + q_1^t) \cdot P^t E_{\tilde{\tau}}^t C_t. \tag{5.4}$$

We refer to this value as the *conditioned expected utility (CEU)* of the auditor. We set the probability of quitting as $\{P^t = 0.186\}_{t \in T}$. This is based on our observations from a 3-month period before data collection in 2017 when a *"break the glass"* [163] real-time warning system was deployed at VUMC to naively warn every access request that triggered alert(s). In addition, we apply the same budget update strategy in real time as introduced in section 5.2, where the consumed budget for the current alert relies on whether a warning is sent to the requestor, the alert type, and the associated signaling scheme.

We compare the CEU of the auditor derived from the Bayesian OSSP to those derived from each of the robust solutions we developed (i.e., the ε-robust Bayesian OSSP and the β-robust Bayesian OSSP) across the 15 evaluation days. Note that in practice the auditor regards each incoming alert as a potential attack. As such, we compute the CEU of the auditor over all alerts.

We investigate the robustness of the results by varying several parameters:

- We consider multiple total auditing budgets $B = \{100, 120, 140\}$.
- We vary the loss of the system due to each walking-away event from non-malicious users. We set $\{C_t = \{-1, -5, -10\}\}_{t \in T}$.
- To investigate how the performance of our solutions varies based on the core parameters that control the degree of robustness, we test a spectrum of values for ε and β, respectively. We consider $\varepsilon = \{0, 50, 100, 200, 400, 800, 1600, 3200\}$ and $\beta = \{0.0, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0\}$.

2). Simulated controlled environment

To systematically investigate the performance of our auditing solutions against attackers with different behavior patterns in terms of their rationality degree, we create a controlled environment based on several key configurations in the real setting. To assign values to $\{E_\tau^t\}_{t \in T}$, we leverage the state at the first triggered alert of the first evaluation day (i.e., Day 42) in the real environment. Specifically, we set $\{E_\tau^t\}_{t \in T}$ as [196.1, 28.9, 141.0, 9.8, 25.4, 15.4, 42.8] for the seven alert types. For the same state, we then solve MIQP (5.1) to derive the probability of being caught for each alert type (i.e., $\{p_1^t + p_0^t\}_{t \in T} =$ [0.083, 0.075, 0.084, 0.129, 0.123, 0.109, 0.173]). Additionally, we set $\{C^t = -1, P^t = 0.186\}_{t \in T}$. Based on these configurations, we simulate the following game components to establish the evaluation environment.

- We apply the derived $\{p_1^t + p_0^t\}_{t \in T}$ to each game, and then introduce random noise to $\{p_1^t + p_0^t\}_{t \in T}$ to simulate games with different budgets. This also bypasses solving $\{B^t\}_{t \in T}$, which fairly simplifies the computation. The noise follows a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. We ensure the simulated probabilities are in the correct range of $(0, 1)$ via value clipping. This process is repeated 2000 times in deriving solutions for all game models.

- Similar to the evaluation in the real environment, we assess performance on a range of values of the two robustness parameters ε and β.

- We simulate attackers with different levels of rationality by controlling the probabilities of deviation from their optimal attacking target. This is achieved by using the *Quantal Response* strategy [153], which assumes that attackers respond stochastically such that the chance of selecting a strategy positively associates with the expected utility on this strategy. Mathematically, the probability of attacking target t for an attacker in type l is:

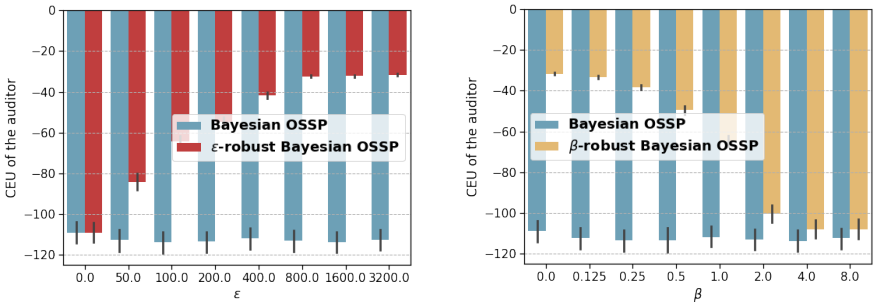
$$w^{t,l} = \frac{e^{\lambda \cdot (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l})}}{\sum_t e^{\lambda \cdot (p_0^t U_{a,c}^{t,l} + q_0^t U_{a,u}^{t,l})}}, \tag{5.5}$$

  where λ (> 0) controls the rationality level of an attacker. A smaller λ makes an attacker more likely to select non-optimal response strategies and vice versa. In the extreme case, a λ value that is infinitely close to 0 leads to an attacker who selects a response strategy in uniformly random manner. We set values for λ as $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$.

5.5.3 Results

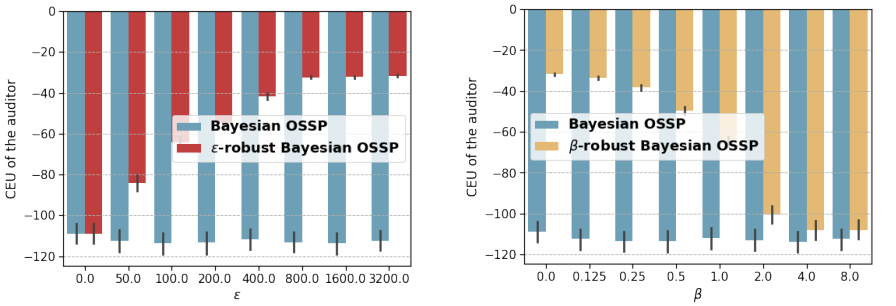
1). Real environment

In Fig. 5.3, we show the performance of the ε-robust Bayesian SAG and the β-robust Bayesian SAG in comparison to the non-robust model (i.e., the Bayesian SAG) across all 15 evaluation days. We present the results under three different choices of total budgets, but apply $\{C_t = -1\}_{t \in T}$ for all scenarios. Due to the fact that the attacker types are randomly assigned for each comparison group, the CEU of the auditor for the Bayesian OSSP across the spectrum of ε and β values differ slightly, but are sufficiently similar for comparison purposes.



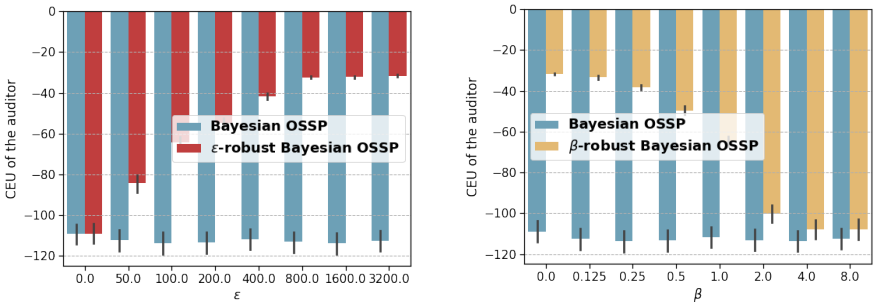
(a) B=100, ε-robust vs non-robust

(b) B=100, β-robust vs non-robust



(c) B=120, ε-robust vs non-robust

(d) B=120, β-robust vs non-robust



(e) B=140, ε-robust vs non-robust

(f) B=140, β-robust vs non-robust

Figure 5.3: CEU of the auditor in the ε-robust and β-robust equilibria compared with the non-robust solution across 15 testing days under different total budgets.

There are several notable findings and implications worth highlighting. First, both the

Table 5.3: Average CEU of the auditor across 15 testing days for $\varepsilon = 3200$ and $\beta = 0.0$. The percentage value in each cell indicates the averaged improvement when compared to the non-robust baseline, i.e., Bayesian OSSP.

B	$\varepsilon = 3200$						$\beta = 0.0$					
	$C_t = -1$		$C_t = -5$		$C_t = -10$		$C_t = -1$		$C_t = -5$		$C_t = -10$	
100	−40.21	70.17%	−92.78	33.96%	−101.80	31.12%	−40.09	70.40%	−94.04	34.68%	−99.94	31.67%
120	−35.62	72.35%	−84.63	35.49%	−95.50	31.84%	−35.60	71.37%	−89.11	35.00%	−93.61	33.85%
140	−31.57	71.93%	−78.11	36.56%	−86.40	34.51%	−31.71	70.91%	−80.91	36.31%	−86.35	33.02%

ε-robust Bayesian OSSP and the β-robust Bayesian OSSP outperform the Bayesian OSSP for almost all values of ε and β by achieving a larger CEU of the auditor, and such an advantage is consistent across all three total budgets. Second, for $\varepsilon = 0$, the ε-robust Bayesian OSSP is as poor as the Bayesian OSSP. Similar observations are made for large values of β (e.g., $\beta = 8.0$ in Figures 5.3b, 5.3d, and 5.3f). These results empirically verify Theorem 10 — both robust solutions degrade to the Bayesian OSSP, which lacks robustness. Third, the CEU of the auditor tends to increase as we increase the total budget from 100 to 140 for every value of ε and β. This is not surprising because a greater amount of protection is realized in auditing.

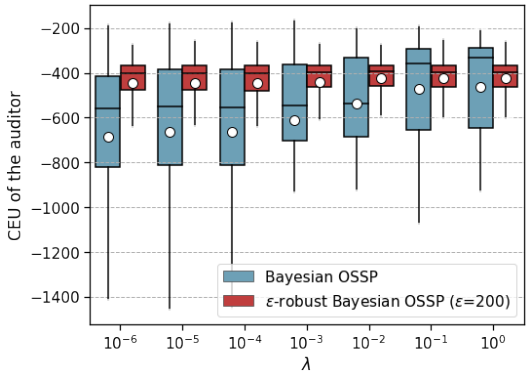
We highlight that the ε-robust Bayesian OSSP for $\varepsilon \geq 800$ and the β-robust Bayesian OSSP for $\beta \leq 0.125$ achieve the highest CEU of the auditor. And, notably, they demonstrate an equally strong capability in the face of real world attackers with respect to the CEU of the auditor. For instance, both of the robust solutions improve the auditing performance by approximately 70% (in terms of the absolute average improvement on the CEU of the auditor divided by the CEU of the auditor from the Bayesian OSSP), which illustrates the effectiveness of our solutions when compared to the state-of-the-art non-robust solution. Interestingly, this phenomenon is in line with the scenario articulated in Theorem 9. As such, the largest improvement from the robust solutions may correspond to a *pseudo-MAXIMIN* solution on this dataset. This implies that the attackers' responses in the real audit logs are almost totally random. One possible explanation for this observation is that the majority of the access requests are not malicious. It is also possible that the cur-

rently deployed auditing strategy (i.e., break the glass) might not help incentivize attackers to behave strategically. And, if this were done well, then it certainly favors the auditor.

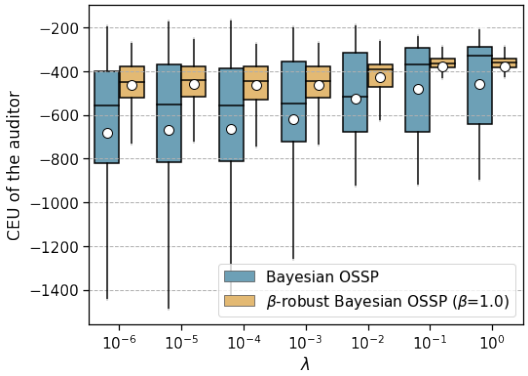
We expanded our investigations to account for multiple different game settings. Given $\varepsilon = 3200$ and $\beta = 0.0$. Table 5.3 illustrates the performance of our robust solutions and their advantages (in terms of average percentage of improvement) over the non-robust solution respectively by varying 1) total budgets for auditing B, and 2) the loss due to each walking-away event of non-malicious data requestor C_t. We selected the values of ε and β that correspond to the best performance exhibited in Fig. 5.3. Here, there are several points to note. First, the ε-robust Bayesian OSSP and the β-robust Bayesian OSSP consistently outperform the Bayesian OSSP with respect to the average CEU of the auditor in a variety of auditing settings. The smallest improvement, which occurs at $\{B = 100, \varepsilon = 3200, \{C_t = -10\}_{t \in T}\}$, over the non-robust solution is still significant and as high as 31%. Second, as C_t increases (i.e., the loss for each walking-away event from non-malicious users decreases), the percentage improvement of our robust solutions also increases. For example, when $B = 100$, the percentage improvement for both robust solutions increases from 31% for $\{C_t = -10\}_{t \in T}$ to 70% for $\{C_t = -1\}_{t \in T}$. This is because the usability cost accounts for more proportions in the CEU of the auditor, which, in turn, reduces the of our solutions' robustness against the imperfectly rational attackers. It should also be noted that the absolute values of the CEU of the auditor monotonically increases with the overall budget B, whereas the percentage improvement remains roughly the same.

2). Simulated controlled environment

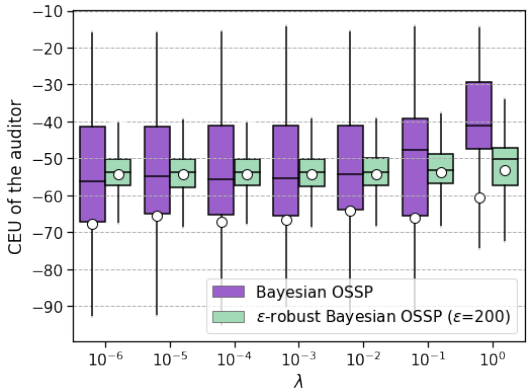
Considering the attacker type as a factor, we compare each robust solution to the Bayesian OSSP across a spectrum of attacker rationality. As shown in Fig. 5.4a and 5.4b, the Bayesian OSSP for F-MOT attackers becomes increasingly poor (the mean CEU of the attacker monotonically decreases) as the attacker's rationality level decreases. By contrast, the robust solutions are relatively stable for different rationality levels, though there is a



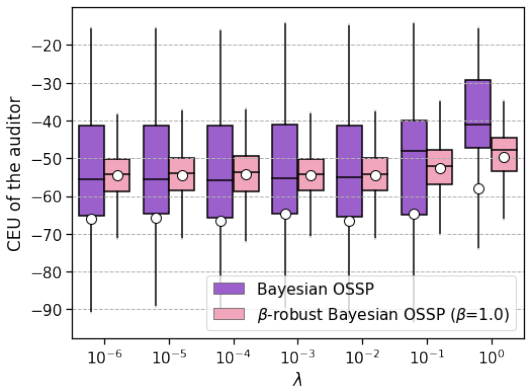
(a) F-MOT attacker in ε-robust SAG



(b) F-MOT attacker in ε-robust SAG



(c) C-MOT attacker in ε-robust SAG



(d) C-MOT attacker in β-robust SAG

Figure 5.4: The difference in the CEU of the auditor in the auditing scenario with Bayesian ε-robust (β-robust) OSSP and the scenario with the alternative Bayesian OSSP ($\varepsilon = 200$ and $\beta = 1.0$). Each box plot indicates the first and third quartiles, as well as the median value. The white circles indicate mean values.

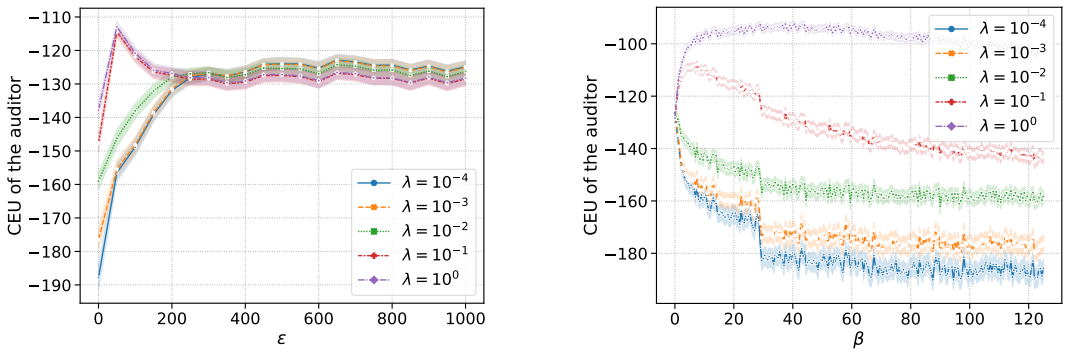
slightly increasing trend for more rational attackers regarding the mean CEU of the auditor. It is also notable that the distributions of auditor's CEU demonstrate larger variances for the Bayesian OSSP than either robust solution, and are left skewed (due to outliers with large negative CEU values). Such biases are much larger for C-MOT attackers (Fig. 5.4c and 5.4d) in that the mean CEU value is even no greater than the corresponding first quartile number. In addition, comparing to the gradual shift of the mean CEU of the auditor over attacker's rationality spectrum for F-MOT attackers in the Bayesian OSSP, this value demonstrates two clear patterns for C-MOT attackers: one for $\lambda = 10^0$ and the other for the rest. The main reason is that the probabilities of selecting strategies for C-MOT attackers is

uniformly distributed across all targets for $\lambda \leq 10^{-1}$. By contrast, F-MOT attackers, whose payoff structures are one magnitude higher than C-MOT attackers, are more responsive to λ values.

Fig. 5.5 shows the CEU of the auditor along ε and β values against attackers with different rationality levels. Again, there are several notable observations. First, for $\varepsilon = 0$, an attacker with a higher level of rationality associates with a higher CEU values of the auditor, as shown in Fig. 5.5a. This is not surprising for several reasons. Specifically, in this scenario, the ε-robust Bayesian OSSP is equivalent to the Bayesian OSSP (see Theorem 10 and Fig. 5.2). Additionally, the solution is derived by assuming a perfectly rational attacker, whose expected utility is optimized. In other words, the auditor can benefit from an increase of the attacker's rationality. Second, the CEU of the auditor against attackers with low rationality (e.g., $\lambda = 10^{-2}, 10^{-3}$, or 10^{-4}) increases with ε in $\varepsilon \in [0, 300]$ and then remains in a relatively stable level for $\varepsilon > 300$. In other words, less rational attackers are better handled by the auditing solutions that account for a large deviation of their utility. By contrast, the CEU of the auditor against attackers with high rationality (e.g., $\lambda = 10^{-1}$ or 10^{0}) demonstrates a clear rise-and-fall pattern before reaching stability. This is because these attackers (who are less likely to choose sub-optimal strategies than other attackers) can be better accommodated by a particular level of deviation in game modeling, which is not necessarily too large. A larger ε renders the model to supply more protection than needed, which thus sacrifices auditor's utility on average, whereas a smaller ε may result in the overshoot of attackers, where unexpected loss can occur to the auditor. Third, it is an interesting observation that in the stable area of the CEU of the auditor ($\varepsilon > 300$), less rational attackers are slightly better handled by the ε-robust Bayesian OSSP. This is because the game shifts the budget to accommodate these attackers.

As expected, the observations above have a similar result for the β-robust Bayesian OSSP solutions as shown in Fig. 5.5b. In general, the findings, as well as implications, from the two subfigures of Fig. 5.5 are in agreement with each other. However, the β-robust

Bayesian OSSP demonstrate inverted patterns along the robustness parameters, which is well explained by the theoretical properties articulated in Section 5.4.



(a) Along ε spectrum.

(b) Along β spectrum.

Figure 5.5: CEU of the auditor with respect to attackers with distinct rationality levels. Each point corresponds to $25K$ simulated game instances.

3). Running time

As the signaling mechanism for auditing functions in real time, it is critical to ensure that an auditing solution is derived quickly. If this is not achieved, the operational efficiency of an organization will be significantly hampered. We measured the time needed to derive the robust solutions using a UNIX server (with two Intel Xeon E5-2650 CPU @ 2.2GHz and 256GB of Memory) for 100 randomly generated game instances of each robust game. We investigate our scalability in deriving robust solutions against a range of alert type numbers, as well as two quantities of attacker types: $|L| = 2$ (the setting in this study) and a much large case where $|L| = 8$. As shown in Fig. 5.6, for our current setting ($|L| = 2, |T| = 7$), the mean running time for both robust solutions are approximately 0.03 seconds, which would be imperceptible. Even for 28 alert types, the mean running time is no greater than 0.1 second. We also observe that deriving the β-robust Bayesian OSSP is more scalable than deriving the ε-robust Bayesian OSSP. In particular, for $|L| = 8, |T| = 28$, it takes 10 seconds to solve the ε-robust Bayesian OSSP, which is 10 times longer than solving the β-robust Bayesian OSSP.

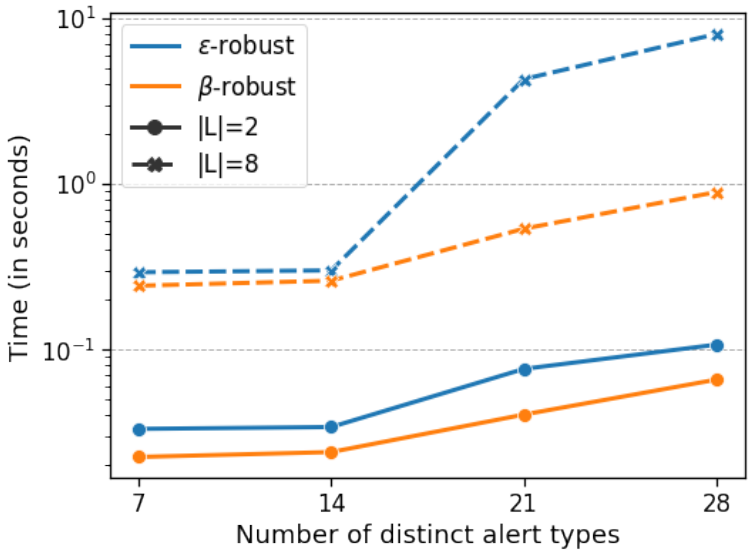


Figure 5.6: Running time for solving the ε- and β-robust Bayesian OSSP.

5.6 Discussion and Conclusion

Alert-based auditing mechanisms for data misuse are widely deployed in database systems. In this paper, we addressed problems with two major assumptions inherent in SAGs, the state-of-the-art database access auditing framework: 1) a single type of attacker (or an identical goal for all attackers) and 2) perfectly rational attackers. To solve the first issue, we leveraged a Bayesian modeling technique to account for different types of attackers by modeling the interactions between the auditor and system users. We then designed two robust signaling auditing models (i.e., the ε- and β-robust Bayesian SAG), as well as corresponding algorithmic strategies to solve these models (i.e., the ε- and β-robust Bayesian OSSP), to address the imperfect rationality of attackers. We demonstrate that our robust solutions largely improve the performance of database auditing using real and controlled environments. Additionally, we found that solving the ε- and β-robust Bayesian SAG can be performed in a running time that is likely to be imperceptible to human. Moreover, the β-robust Bayesian SAG demonstrates higher scalability for larger sized auditing profiles (in terms of the number of attacker types and alert types).

Still, there are several limitations to point out for future investigations. First, the EHR

access dataset used in the real environment was not based on any explicit game theoretic implementation of auditing. As a consequence, this dataset is likely to be representative of certain values of ε and β already. Despite this fact, we believe this dataset is useful for evaluation purposes because it illustrates the performance of different solutions and provides a surrogate for a real system. Still, it should be recognized that EHR users (and especially attackers) may slowly evolve their practices to influence the system, which may require amendments to our solutions. Second, though we conducted an evaluation on variety of game profiles, we did not study how to tune the values of ε or β in practice. Though we consider this to be outside the scope of this study, it is critical to optimize the deployment of these parameters in real world settings. Third, we assume that alerts are triggered independently along the timeline, such that attackers do not collaborate. This may not always be the case, in which case future investigations should consider how to consider strategic coordination of adversarial groups.

Part IV

Conclusion

Chapter 6

Conclusions and Future Directions

6.1 Summary

This dissertation focused on solving a challenge threat to personal privacy encountered by almost all modern information systems, the auditing of suspicious access to data. The methodology developed throughout this dissertation is based on an understanding of the factors inherent in the current auditing process, which are hallmarked by 1) a large amount of triggered alerts that are substantially beyond the available auditing budget, 2) a high rate of false positives, 3) highly strategic attackers who can evade carelessly designed defense strategy, and 4) the dynamic nature of the data access events that requires investigation. Game theoretic modeling addresses these factors, and, thus, is a natural approach support administrators who need to perform auditing. The developed mechanisms and their solutions enable a more effective and efficient auditing than our current status quo. The results of our empirical investigations are remarkable because they demonstrate that blending an economic perspective and technical approaches can dramatically improve the system administrator's auditing capability in a budget-constrained adversarial environment. Moreover, our auditing frameworks provides for an explicit attacker deterrence mechanism while maximizing its effect through strategy randomization and signaling. The specific accomplishments and innovations made through this dissertation are summarized as follows.

Accomplishment 1. Designed and optimized the alert type prioritization and budget allocation to maximize audit effectiveness. We introduced a novel formalization of the type-based alert prioritization and budget allocation as a Stackelberg security game between an auditor and a set of attackers. In this game framework, the auditor chooses an alert prioritization policy (i.e., the probability distribution over the permutation of alert

types), whereas attackers determine their nature of violations, or are deterred in response. Our research illustrates that data misuse auditing, as a significant component of system management, can be improved by prioritizing which alerts to focus on via a game theoretic framework, allowing auditing policies to make the best use of limited auditing resources while simultaneously accounting for the strategic behavior of potential policy violators.

Accomplishment 2. Incorporated real time information exchange between players and made it an advantage of the auditor to influence attackers. We extended the strategic modeling advantage to the real time environment. Specifically, we incorporated a concept—online signaling—into game formalization through embedding a warning mechanism (e.g., via a private message box) into the gap between access request made by the potential attacker and the actual execution of the attack. The resulting game framework, i.e., SAG, is designed to optimize the warning strategy and the audit decision for each incoming alert. The key constraint in this game enables the valid deterrence of attackers because it forces that the best strategy for an attacker when receiving a warning message is to quit the current request. The most prominent finding is that the information advantage of the auditor can be translated into the gain of their expected utility in the data misuse auditing scenario. And this is achieved by the termination of a portion of ongoing attacks by themselves.

Accomplishment 3. Addressed the practical adversarial environment where attackers have diverse goals (or types) and imperfect rationality. To bridge the gap between the oversimplified assumptions of the SAG and its application domain, we developed a new game framework, i.e., robust Bayesian SAG, which considers attackers with diverse goals in the system and imperfect rationality. To do so, we explicitly modeled the auditor's uncertainty about the goal of the encountered attackers and introduce two distinct methods to bound players' deviations from their corresponding optimal strategies. We empirically showed that our solutions largely improve the performance of data misuse auditing in the real world and can handle any attackers in the rationality spectrum.

Notably, the game theoretic frameworks in this dissertation demonstrate advantages in generalizability and scalability. First, our auditing frameworks are inclusive because they enable embedding any potential malicious situations and their corresponding detection methods or rules as alert types. In other words, as long as the adversarial behaviors of attackers can be represented by patterns, our framework can include them in deriving auditing solutions, no matter how powerful an attacker could be. Second, according to the scalability investigations in this dissertation, solving the developed audit games of a large game size takes acceptable time even in a real time setting. For example, the running time of a robust Bayesian SAG with 28 alert types is imperceptible to humans, which will induce negligible burden to normal data accesses. Third, to the best of our knowledge, the "break the glass" policy has been integrated as a signaling function in many mission-critical information system (such as Epic EHR system), such that the testing and deployment of our online signaling solutions can benefit from these engineering efforts.

6.2 Future Investigations

Moving forward, there are several limitations of tis work that can serve as opportunities for further investigation.

> **Open Question 1:** *How should we model the imprecise knowledge of game players when deriving their strategies in audit games? Can such a phenomenon substantially influence the auditing performance?*

The game theoretic auditing frameworks developed in this dissertation assume that players know precisely what they need in terms of prior knowledge to compute their own expected utility in each possible situation. For example, in all of the three leader-follower games introduced in this body of work, we assumed an attacker is aware of the knowledge of the auditor's defense policy, which is typically a probability distribution over a finite action space. In the literature of the security games, it is typically believed that this can

be obtained through careful observations or analysis of historical data. However, such approaches are unlikely to yield highly accurate approximations of the necessary information in practice. In particular, the signaling scheme of the SAG is derived in real time, and, thus, dynamic over time. And it is evident that the temporal characteristic of the framework increases the complexity for an attacker to learn the accurate signaling scheme. Although our robust Bayesian SAG framework can partially reduce the influence of this factor, a systematic investigation is needed to assess its influence for the overall auditing performance.

> **Open Question 2:** *How can we determine the robustness level of the auditing frameworks?*

Chapter 5 demonstrates that both of our robust solutions are able to handle attackers when their rationality levels range from perfect rational to totally irrational. Given the rationality level of an attacker, there exists a particular set of values for the key robust parameters (i.e., ε and β) that can maximize the auditor's expected utility, while other values may lead to worse auditing performance. As a consequence, it is important to ensure that the derived auditing solutions match the rational level of the real world attackers. Yet this is a non-trivial problem because 1) verified real attacks are quite rare in practice, such that there is limited information that can be leaned upon to learn an accurate approximation of attacker's level of rationality and 2) an attacker's pattern of behavior can evolve over time. As such, it evident that our methods would benefit from the incorporation of behavioral science to analyze, monitor and model this problem.

> **Open Question 3:** *How can we harmonize different data misuse auditing or detection frameworks to maximize the overall auditing performance can be maximized?*

In this dissertation, we investigated how an alert prioritization policy and an online signaling strategy can be designed to improve data misuse auditing in a separate manner.

These two approaches depart from different perspectives, and benefit the auditing task in their own way. There is no clear guidance on when to use which in practice. However, it is natural that new auditing or data misuse detection frameworks in other novel perspectives (which does not necessarily adopt game theoretic frameworks) will be developed. And these approaches will not necessarily conflict with, or dominate, one another. Intuitively, the harmonization of multiple auditing approaches from different perspectives can enable the construction of a more protective environment. For example, a real time scoring system based on machine learning can be embedded into the robust Bayesian SAG by adding new alert types. However, this would cause new issues to arise, such as how to assign reasonable payoff structures for new alert types which do not correspond to explainable motivation of an attacker. Overall, this issue remains largely unexplored and needs systematic investigations.

Open Question 4: *How can we transition the developed auditing frameworks into real world deployment?*

In this dissertation, we depart from a computational perspective and demonstrate the potentiality of the developed auditing frameworks using well controlled evaluation environments. The ultimate goal of this line of research should be to deploy our methodology into the real-world auditing environment and guarantee their efficacy and efficiency along the timeline; however, there exist multiple challenges beyond the computational solutions that this dissertation contributes. First, we need to investigate how to embed our frameworks (more specifically, the recommendations of which to audit) into the existing auditing workflow such that the synergy of experts and AI can be optimized given limited auditing budgets. This usually intertwines with the issue of trust, which needs to be established via comprehensive evaluation and communication. Second, it is very important to investigate the corresponding legal concern. Given the uncertain nature of our auditing policies, it is possible that our solutions neither warn a malicious data access nor investigate it at the end of an audit cycle. Though the overall auditing performance is improved, the victims of

data breach in the mentioned scenario may question that their data was not well protected compared to others.

> **Open Question 5:** *Can data synthesis (or augmentation) be leveraged to enhance the performance of auditing and support model reproducibility via public data dissemination?*

We recognize that the study of access auditing and the development of tools to detect inappropriate access is currently limited to those with access to specific access log data. There is a need to democratize this field of study by developing tools to generate realistic access data, with the associated clinical context and support reproducibility. The release of simulated, yet realistic access data will allow for the broader community to compare methods and results with a common playing field, and meanwhile mitigate the concerns (such as privacy) from publishing the raw data. Also, there is mounting evidence that the performance of AI models can be improved by incorporating simulated data that is representative of the space. Our experience on EHR synthesis [164, 165, 166] shows that such simulation is possible. However, there remain challenges to adapting the method to generate access logs. One challenge is that access logs contain various data types, which leads to a substantially larger feature space. The other is that access logs have temporal dependencies (access trajectory) that need to be represented.

BIBLIOGRAPHY

[1] Jorge Tavares and Tiago Oliveira. Electronic health record patient portal adoption by health care consumers: An acceptance model and survey. *Journal of Medical Internet Research*, 18(3), 2016.

[2] Raymond Tsai, Elijah J Bell III, Hawkin Woo, Kevin Baldwin, and Michael A Pfeffer. How patients use a patient portal: an institutional case study of demographics and usage patterns. *Applied Clinical Informatics*, 10(1):96, 2019.

[3] Vitaly Herasevich, John Litell, and Brian Pickering. Electronic medical records and mhealth anytime, anywhere. *Biomedical Instrumentation & Technology*, 46(s2):45–48, 2012.

[4] Blackford Middleton, Meryl Bloomrosen, Mark A Dente, Bill Hashmat, Ross Koppel, J Marc Overhage, Thomas H Payne, S Trent Rosenbloom, Charlotte Weaver, and Jiajie Zhang. Enhancing patient safety and quality of care by improving the usability of electronic health record systems: recommendations from amia. *Journal of the American Medical Informatics Association*, 20(e1):e2–e8, 2013.

[5] Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. *Risk Management and Healthcare Policy*, 4:47, 2011.

[6] Cem Dener, Joanna Watkins, and William Leslie Dorotinsky. *Financial management information systems: 25 years of World Bank experience on what works and what doesn't*. The World Bank, 2011.

[7] Liu Liu, Olivier De Vel, Qing-Long Han, Jun Zhang, and Yang Xiang. Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, 20(2):1397–1417, 2018.

[8] George Silowash, Dawn Cappelli, Andrew Moore, Randall Trzeciak, Timothy J. Shimeall, and Lori Flynn. Common sense guide to mitigating insider threats. 2012.

[9] Daniel L Costa, Michael J Albrethsen, and Matthew L Collins. Insider threat indicator ontology. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA PITTSBURGH United States, 2016.

[10] Reed Abelson and Matthew Goldstein. Millions of anthem customers targeted in cyberattack. https://www.nytimes.com/2015/02/05/business/hackers-breached-data-of-millions-insurer-says.html, 2015.

[11] Sara Ashley O'Brien. Giant equifax data breach: 143 million people could be affected. http://money.cnn.com/2017/09/07/technology/business/equifax-data-breach/index.html, 2017.

[12] Radha Jagadeesan, Alan Jeffrey, Corin Pitcher, and James Riely. Towards a theory of accountability and audit. In *Proceedings of the 2009 European Symposium on Research in Computer Security*, pages 152–167, 2009.

[13] Ryan KL Ko, Bu Sung Lee, and Siani Pearson. Towards achieving accountability, auditability and trust in cloud computing. In *Proceedings of the 2011 International Conference on Advances in Computing and Communications*, pages 432–444, 2011.

[14] Brent R Waters, Dirk Balfanz, Glenn Durfee, and Diana K Smetters. Building an encrypted and searchable audit log. In *Proceedings of the 2004 Network and Distributed System Security Symposium*, volume 4, pages 5–6, 2004.

[15] Marcello Cinque, Raffaele Della Corte, and Antonio Pecchia. Contextual filtering and prioritization of computer application logs for security situational awareness. *Future Generation Computer Systems*, 111:668–680, 2020.

[16] Xinmeng Zhang, Chao Yan, Bradley A Malin, Mayur B Patel, and You Chen. Predicting next-day discharge via electronic health record access logs. *Journal of the American Medical Informatics Association*, 28(12):2670–2680, 2021.

[17] Chao Yan, Xinmeng Zhang, Cheng Gao, Erin Wilfong, Jonathan Casey, Daniel France, Yang Gong, Mayur Patel, Bradley Malin, and You Chen. Collaboration structures in covid-19 critical care: Retrospective network analysis study. *JMIR Human Factors*, 8(1):e25724, 2021.

[18] Hannah Mannering, Chao Yan, Yang Gong, Mhd Wael Alrifai, Daniel France, You Chen, et al. Assessing neonatal intensive care unit structures and outcomes before and during the covid-19 pandemic: Network analysis study. *Journal of medical Internet research*, 23(10):e27261, 2021.

[19] You Chen, Chao Yan, and Mayur B Patel. Network analysis subtleties in icu structures and outcomes. *American Journal of Respiratory and Critical Care Medicine*, 202(11):1606–1607, 2020.

[20] Konstantin Berlin, David Slater, and Joshua Saxe. Malicious behavior detection using windows audit logs. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pages 35–44, 2015.

[21] Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security (TISSEC)*, 2(2):159–176, 1999.

[22] Andrew Sutton and Reza Samavi. Blockchain enabled privacy audit logs. In *Proceedings of the 2017 International Semantic Web Conference*, pages 645–660, 2017.

[23] José Luis Fernández-Alemán, Inmaculada Carrión Señor, Pedro Ángel Oliver Lozoya, and Ambrosio Toval. Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, 46(3):541–562, 2013.

[24] Julia Adler-Milstein, Jason S Adelman, Ming Tai-Seale, Vimla L Patel, and Chris Dymek. Ehr audit logs: a new goldmine for health services research? *Journal of biomedical informatics*, 101:103343, 2020.

[25] Hanna Mazzawi, Gal Dalal, David Rozenblatz, Liat Ein-Dorx, Matan Niniox, and Ofer Lavi. Anomaly detection in large databases using behavioral patterning. In *Proceedings of the 33rd IEEE International Conference on Data Engineering*, pages 1140–1149, 2017.

[26] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336, 2013.

[27] Sandeep Bhatt, Pratyusa K Manadhata, and Loai Zomlot. The operational role of security information and event management systems. *IEEE security & Privacy*, 12(5):35–41, 2014.

[28] Neminath Hubballi and Vinoth Suryanarayanan. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, 49:1–17, 2014.

[29] Richard Zuech, Taghi M Khoshgoftaar, and Randall Wald. Intrusion detection and big heterogeneous data: a survey. *Journal of Big Data*, 2(1):1–41, 2015.

[30] Huseyin Cavusoglu, Birendra Mishra, and Srinivasan Raghunathan. A model for evaluating it security investments. *Communications of the ACM*, 47(7):87–92, 2004.

[31] Charles Feng, Shuning Wu, and Ningwei Liu. A user-centric machine learning framework for cyber security operations center. In *Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics*, pages 173–175, 2017.

[32] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2015.

[33] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):1–39, 2013.

[34] Cuong T Do, Nguyen H Tran, Choongseon Hong, Charles A Kamhoua, Kevin A Kwiat, Erik Blasch, Shaolei Ren, Niki Pissinou, and Sundaraja Sitharama Iyengar. Game theory for cyber security and privacy. *ACM Computing Surveys*, 50(2):30, 2017.

[35] Shui Yu, Guojun Wang, Xiting Liu, and Jianwei Niu. Security and privacy in the age of the smart internet of things: An overview from a networking perspective. *IEEE Communications Magazine*, 56(9):14–18, 2018.

[36] Chao Yan, Bo Li, Yevgeniy Vorobeychik, Aron Laszka, Daniel Fabbri, and Bradley Malin. Get your workload in order: Game theoretic prioritization of database auditing. In *Proceedings of the 34th IEEE International Conference on Data Engineering*, pages 1304–1307, 2018.

[37] Chao Yan, Bo Li, Yevgeniy Vorobeychik, Aron Laszka, Daniel Fabbri, and Bradley Malin. Database audit workload prioritization via game theory. *ACM Transactions on Privacy and Security*, 22(3):17, 2019.

[38] Chao Yan, Haifeng Xu, Yevgeniy Vorobeychik, Bo Li, Daniel Fabbri, and Bradley A Malin. To warn or not to warn: Online signaling in audit games. In *Proceedings of the 36th IEEE International Conference on Data Engineering*, pages 481–492, 2020.

[39] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

[40] Paul John Steinbart, Robyn L Raschke, Graham Gal, and William N Dilla. The influence of a good relationship between the internal audit and information security functions on information security outcomes. *Accounting, Organizations and Society*, 71:15–29, 2018.

[41] Tahera Yesmin and Michael W Carter. Evaluation framework for automatic privacy auditing tools for hospital data breach detections: A case study. *International journal of medical informatics*, 138:104123, 2020.

[42] Qiaona Hu, Baoming Tang, and Derek Lin. Anomalous user activity detection in enterprise multi-source logs. In *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops*, pages 797–803, 2017.

[43] Sanket Shah, Arunesh Sinha, Pradeep Varakantham, Andrew Perrault, and Milind Tambe. Solving online threat screening games using constrained action space reinforcement learning. *arXiv preprint arXiv:1911.08799*, 2019.

[44] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[45] Sunu Mathew, Michalis Petropoulos, Hung Q Ngo, and Shambhu J Upadhyaya. A data-centric approach to insider attack detection in database systems. In *Proceedings of the 2010 International Symposium on Research in Attacks, Intrusions and Defenses*, pages 382–401, 2010.

[46] Ashish Kamra and Elisa Ber. Survey of machine learning methods for database security. *Machine Learning in Cyber Trust*, pages 53–71, 2009.

[47] Ashish Kamra, Evimaria Terzi, and Elisa Bertino. Detecting anomalous access patterns in relational databases. *International Journal on Very Large Data Bases*, 17(5):1063–1077, 2008.

[48] Sarah Heckman and Laurie Williams. On establishing a benchmark for evaluating static analysis alert prioritization and classification techniques. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 41–50, 2008.

[49] Amreen Sultana and MA Jabbar. Intelligent network intrusion detection system using data mining techniques. In *Proceedings of the 2016 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 329–333, 2016.

[50] Elham Ariafar and Rasoul Kiani. Intrusion detection system using an optimized framework based on datamining techniques. In *Proceedings of the 4th IEEE International Conference on Knowledge-Based Engineering and Innovation*, pages 0785–0791, 2017.

[51] William R Cook and Martin R Gannholm. Rule based database security system and method, November 16 2004. US Patent 6,820,082.

[52] Sriram Samu, Namit Jain, and Wei Wang. Database system event triggers, June 11 2002. US Patent 6,405,212.

[53] Ron Ben-Natan. System and methods for nonintrusive database security, October 14 2008. US Patent 7,437,362.

[54] Aziz A Boxwala, Jihoon Kim, Janice M Grillo, and Lucila Ohno-Machado. Using statistical and machine learning to help institutions detect suspicious access to electronic health records. *Journal of the American Medical Informatics Association*, 18(4):498–505, 2011.

[55] You Chen, Steve Nyemba, and Bradley Malin. Detecting anomalous insiders in collaborative information systems. *IEEE Transactions on Dependable and Secure Computing*, (99):1–1, 2012.

[56] You Chen, Steve Nyemba, Wen Zhang, Bradley Malin, HY Shahir, U Glässer, R Farahbod, P Jackson, H Wehn, K Glass, et al. Specializing network analysis to detect anomalous insider actions. *Security Informatics*, 1(1):5, 2012.

[57] Daniel Fabbri and Kristen LeFevre. Explaining accesses to electronic medical records using diagnosis information. *Journal of the American Medical Informatics Association*, 20(1):52–60, 2013.

[58] Daniel Fabbri, Ravi Ramamurthy, and Raghav Kaushik. Select triggers for data auditing. In *Proceedings of the 29th IEEE International Conference on Data Engineering*, pages 1141–1152, 2013.

[59] Daniel Fabbri and Kristen LeFevre. Explanation-based auditing. *Proceedings of the 2011 VLDB Endowment*, 5(1):1–12, 2011.

[60] EWT Ngai, Yong Hu, YH Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature. *Decision Support Systems*, 50(3):559–569, 2011.

[61] Linda Delamaire, HAH Abdou, and John Pointon. Credit card fraud and detection techniques: A review. *Banks and Bank Systems*, 4(2):57–68, 2009.

[62] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

[63] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun Majumdar. Credit card

fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008.

[64] Philip K Chan, Wei Fan, Andreas L Prodromidis, and Salvatore J Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and their Applications*, 14(6):67–74, 1999.

[65] Rong-Chang Chen, Tung-Shou Chen, and Chih-Chiang Lin. A new binary support vector system for increasing detection rate of credit card fraud. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(02):227–239, 2006.

[66] R Brause, T Langsdorf, and Michael Hepp. Neural data mining for credit card fraud detection. In *Proceedings of the International Conference on Tools with Artificial Intelligence*, pages 103–106, 1999.

[67] Michael Sternberg and Robert G. Reynolds. Using cultural algorithms to support re-engineering of rule-based expert systems in dynamic performance environments: a case study in fraud detection. *IEEE Transactions on Evolutionary Computation*, 1(4):225–243, 1997.

[68] Mubeena Syeda, Yan-Qing Zhang, and Yi Pan. Parallel granular neural networks for fast credit card fraud detection. In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, pages 572–577, 2002.

[69] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

[70] Ali Ahmadian Ramaki, Abbas Rasoolzadegan, and Abbas Ghaemi Bafghi. A systematic mapping study on intrusion alert analysis in intrusion detection systems. *ACM Computing Surveys (CSUR)*, 51(3):1–41, 2018.

[71] Mansoor Nasir, Khan Muhammad, Paolo Bellavista, Mi Young Lee, and Muhammad Sajjad. Prioritization and alert fusion in distributed iot sensors using kademlia based distributed hash tables. *IEEE Access*, 8:175194–175204, 2020.

[72] Yuxin Meng and Lam-For Kwok. Enhancing false alarm reduction using voted ensemble selection in intrusion detection. *International Journal of Computational Intelligence Systems*, 6(4):626–638, 2013.

[73] Sean Carlisto De Alvarenga, Sylvio Barbon Jr, Rodrigo Sanches Miani, Michel Cukier, and Bruno Bogaz Zarpelão. Process mining and hierarchical clustering to help intrusion alert visualization. *Computers & Security*, 73:474–491, 2018.

[74] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In *Proceedings of the 2017 AAAI Conference on Artificial Intelligence Workshop*, 2017.

[75] Frédéric Cuppens and Alexandre Miege. Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 202–215, 2002.

[76] Shisong Xiao, Yugang Zhang, Xuejiao Liu, and Jingju Gao. Alert fusion based on cluster and correlation analysis. In *Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology*, pages 163–168, 2008.

[77] Federico Maggi, Matteo Matteucci, and Stefano Zanero. Reducing false positives in anomaly detectors through fuzzy alert aggregation. *Information Fusion*, 10(4):300–311, 2009.

[78] Khalid Alsubhi, Ehab Al-Shaer, and Raouf Boutaba. Alert prioritization in intrusion detection systems. In *Proceedings of the 2008 IEEE Network Operations and Management Symposium*, pages 33–40, 2008.

[79] Khalid Alsubhi, Issam Aib, and Raouf Boutaba. Fuzmet: A fuzzy-logic based alert prioritization engine for intrusion detection systems. *International Journal of Network Management*, 22(4):263–284, 2012.

[80] Humphrey Waita Njogu and Luo Jiawei. Using alert cluster to reduce ids alerts. In *Proceedings of the 2010 International Conference on Computer Science and Information Technology*, volume 5, pages 467–471, 2010.

[81] Muhamad Erza Aminanto, Lei Zhu, Tao Ban, Ryoichi Isawa, Takeshi Takahashi, and Daisuke Inoue. Automated threat-alert screening for battling alert fatigue with temporal isolation forest. In *Proceedings of the 2019 International Conference on Privacy, Security and Trust*, pages 1–3, 2019.

[82] Muhamad Erza Aminanto, Tao Ban, Ryoichi Isawa, Takeshi Takahashi, and Daisuke Inoue. Threat alert prioritization using isolation forest and stacked auto encoder with day-forward-chaining analysis. *IEEE Access*, 8:217977–217986, 2020.

[83] Muhamad Erza Aminanto, Lei Zhu, Tao Ban, Ryoichi Isawa, Takeshi Takahashi, and Daisuke Inoue. Combating threat-alert fatigue with online anomaly detection using isolation forest. In *Proceedings of the 2019 International Conference on Neural Information Processing*, pages 756–765, 2019.

[84] Li Sun, Steven Versteeg, Serdar Boztas, and Asha Rao. Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study. *arXiv preprint arXiv:1609.06676*, 2016.

[85] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of the 2009 International Conference on Autonomous Agents and Multi-agent Systems*, pages 689–696, 2009.

[86] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: looking beyond a decade of success. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5494–5501, 2018.

[87] Bo An, Milind Tambe, and Arunesh Sinha. Stackelberg security games (ssg): Basics and application overview. *Improving Homeland Security Decisions*, page 485, 2017.

[88] Sixie Yu, Yevgeniy Vorobeychik, and Scott Alfeld. Adversarial classification on social networks. In *Proceedings of the 2018 International Conference on Autonomous Agents and Multi-agent Systems*, pages 211–219, 2018.

[89] Liang Tong, Sixie Yu, Scott Alfeld, and Yevgeniy Vorobeychik. Adversarial regression with multiple learners. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4946–4954, 2018.

[90] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 2006 ACM Conference on Electronic Commerce*, pages 82–90, 2006.

[91] Bernhard Von Stengel and Shmuel Zamir. Leadership with commitment to mixed strategies. Technical report, Citeseer, 2004.

[92] James Pita, Manish Jain, Fernando Ordónez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for los angeles airport security. *AI magazine*, 30(1):43–43, 2009.

[93] Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 425–431, 2016.

[94] Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordonez, and Milind Tambe. Iris-a tool for strategic security allocation in transportation networks. *Proceedings of the 2009 International Conference on Autonomous Agents and Multi-agent Systems*, pages 37–44, 2009.

[95] Bo An, Fernando Ordóñez, Milind Tambe, Eric Shieh, Rong Yang, Craig Baldwin, Joseph DiRenzo III, Kathryn Moretti, Ben Maule, and Garrett Meyer. A deployed quantal response-based patrol planning system for the us coast guard. *Interfaces*, 43(5):400–420, 2013.

[96] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Brian C Schwedock, Milind Tambe, and Andrew Lemieux. PAWS – a deployed game-theoretic application to combat poaching. *AI Magazine*, 38(1):23–36, 2017.

[97] Jeremiah Blocki, Nicolas Christin, Anupam Datta, and Arunesh Sinha. Audit mechanisms for provable risk management and accountable data governance. In *Proceedings of the 2012 International Conference on Decision and Game Theory for Security*, pages 38–59, 2012.

[98] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel D Procaccia, and Arunesh Sinha. Audit games. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 41–47, 2013.

[99] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel D Procaccia, and Arunesh Sinha. Audit games with multiple defender resources. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, volume 15, pages 791–797, 2015.

[100] Rajesh Ganesan, Sushil Jajodia, Ankit Shah, and Hasan Cam. Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):4, 2016.

[101] Aaron Schlenker, Haifeng Xu, Mina Guirguis, Chris Kiekintveld, Arunesh Sinha, Milind Tambe, Solomon Sonya, Darryl Balderas, and Noah Dunstatter. Don't bury your head in warnings: a game-theoretic approach for intelligent allocation of cyber-security alerts. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 381–387, 2017.

[102] Aron Laszka, Yevgeniy Vorobeychik, Daniel Fabbri, Chao Yan, and Bradley Malin. A game-theoretic approach for alert prioritization. In *Proceedings of the AAAI-17 Workshop on Artificial Intelligence for Cyber Security*, 2017.

[103] Liang Tong, Aron Laszka, Chao Yan, Ning Zhang, and Yevgeniy Vorobeychik. Finding needles in a moving haystack: Prioritizing alerts with adversarial reinforcement learning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[104] Zinovi Rabinovich, Albert Xin Jiang, Manish Jain, and Haifeng Xu. Information disclosure as a means to security. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multi-agent Systems*, pages 645–653, 2015.

[105] Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Milind Tambe, Phebe Vayanos, Fei Fang, Long Tran-Thanh, and Yevgeniy Vorobeychik. Deceiving cyber adversaries: A game theoretic approach. In *Proceedings of the 2018 International Conference on Autonomous Agents and Multi-agent Systems*, 2018.

[106] Haifeng Xu, Kai Wang, Phebe Vayanos, and Milind Tambe. Strategic coordination of human patrollers and mobile sensors with signaling for security games. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[107] Haifeng Xu, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe. Exploring information asymmetry in two-stage security games. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1057–1063, 2015.

[108] Shaddin Dughmi and Haifeng Xu. Algorithmic bayesian persuasion. In *Proceedings of the 48th Annual ACM symposium on Theory of Computing*, pages 412–425, 2016.

[109] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. Improving resource allocation strategy against human adversaries in security games. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 22, page 458, 2011.

[110] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the 2013 AAAI Conference on Artificial Intelligence*, 2013.

[111] Andrew McAfee and Erik Brynjolfsson. Big data: The management revolution. *Harvard Business Review*, Oct:3–9, 2012.

[112] Lillian Ablon, Martin C. Libicki, and Andrea A. Golay. *Markets for cybercrime tools and stolen data: hackers' bazaar*. 2014.

[113] Chee-Wooi Ten, Govindarasu Manimaran, and Chen-Ching Liu. Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(4):853–865, 2010.

[114] Horacio D Kuna, Ramón García-Martinez, and Francisco R Villatoro. Outlier detection in audit logs for application systems. *Information Systems*, 44:22–33, 2014.

[115] Carl Gunter, David Liebovitz, and Bradley Malin. Experience-based access management. *IEEE Security and Privacy Magazine*, 9:48–55, 2011.

[116] Rakesh Agrawal and Chris Johnson. Securing electronic health records without impeding the flow of information. *International Journal of Medical Informatics*, 76(5-6):471–479, 2007.

[117] Lillian Rostad and Ole Edsberg. A study of access control requirements for healthcare systems based on audit trails from access logs. In *Proceedings of the 2006 Annual Computer Security Applications Conference*, pages 175–186, 2006.

[118] Peter Kieseberg, Bernd Malle, Peter Frühwirt, Edgar Weippl, and Andreas Holzinger. A tamper-proof audit and control system for the doctor in the loop. *Brain Informatics*, 3(4):269–279, 2016.

[119] Vincent Conitzer. On stackelberg mixed strategies. *Synthese*, 193(3):689–703, 2016.

[120] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard Business Review*, 90(10):60–68, 2012.

[121] Shen Yin and Okyay Kaynak. Big data for modern industry: challenges and trends [point of view]. *Proceedings of the IEEE*, 103(2):143–146, 2015.

[122] Kyriacos E Pavlou and Richard T Snodgrass. Dragoon: An information accountability system for high-performance databases. In *Proceedings of the 28th IEEE International Conference on Data Engineering*, pages 1329–1332, 2012.

[123] Duygu Sinanc Terzi, Ramazan Terzi, and Seref Sagiroglu. A survey on security and privacy issues in big data. In *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions*, pages 202–207, 2015.

[124] Ragib Hasan, Shams Zawoad, Shahid Noor, Md Munirul Haque, and Darrell Burke. How secure is the healthcare network from insider attacks? an audit guideline for vulnerability analysis. In *Proceedings of the 40th IEEE Annual Computer Software and Applications Conference*, volume 1, pages 417–422, 2016.

[125] Mamta Puppala, Tiancheng He, Xiaohui Yu, Shenyi Chen, Richard Ogunti, and Stephen TC Wong. Data security and privacy management in healthcare applications

and clinical data warehouse environment. In *Proceedings of the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pages 5–8, 2016.

[126] Rajeev Motwani, Shubha U Nabar, and Dilys Thomas. Auditing sql queries. In *Proceedings of the 24th IEEE International Conference on Data Engineering*, pages 287–296, 2008.

[127] Wentian Lu and Gerome Miklau. Auditing a database under retention restrictions. In *Proceedings of the 25th IEEE International Conference on Data Engineering*, pages 42–53, 2009.

[128] S Michael Groomer and Uday S Murthy. Continuous auditing of database applications: An embedded audit module approach. In *Continuous Auditing: Theory and Application*, pages 105–124. 2018.

[129] Monica Hedda, Bradley Malin, Chao Yan, and Daniel Fabbri. Evaluating the effectiveness of auditing rules for electronic health record systems. In *Proceedings of the 2017 AMIA Annual Symposium*, volume 2017, pages 866–875, 2017.

[130] Adam Barth, John Mitchell, Anupam Datta, and Sharada Sundaram. Privacy and utility in business processes. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium*, pages 279–294, 2007.

[131] Aron Laszka, Yevgeniy Vorobeychik, Daniel Fabbri, Chao Yan, and Bradley Malin. A game-theoretic approach for alert prioritization. In *Proceedings of the 31st AAAI Workshop on Artificial Intelligence for Cyber Security*, 2017.

[132] Fei Fang, Thanh Hong Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, Andrew Lemieux, et al. Deploying paws: Field optimization of the protection assistant for wildlife security. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.

[133] Zhiyu Wan, Yevgeniy Vorobeychik, Weiyi Xia, Ellen Wright Clayton, Murat Kantarcioglu, Ranjit Ganta, Raymond Heatherly, and Bradley A Malin. A game theoretic framework for analyzing re-identification risk. *PloS One*, 10(3):e0120592, 2015.

[134] Zhiyu Wan, Yevgeniy Vorobeychik, Weiyi Xia, Ellen Wright Clayton, Murat Kantarcioglu, and Bradley Malin. Expanding access to large-scale genomic data while promoting privacy: a game theoretic approach. *The American Journal of Human Genetics*, 100(2):316–322, 2017.

[135] Zhiyu Wan, Yevgeniy Vorobeychik, Weiyi Xia, Yongtai Liu, Myrna Wooders, Jia Guo, Zhijun Yin, Ellen Wright Clayton, Murat Kantarcioglu, and Bradley A Malin. Using game theory to thwart multistage privacy intrusions when sharing data. *Science Advances*, 7(50):eabe9986, 2021.

[136] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Milind Tambe, and Fernando Ordónez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40(4):267–290, 2010.

[137] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[138] Carl A Gunter, David Liebovitz, and Bradley Malin. Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE Security & Privacy*, 9(5):48, 2011.

[139] Vicenç Torra. *Data privacy: Foundations, new developments and the big data challenge*. Springer, 2017.

[140] Abid Mehmood, Iynkaran Natgunanathan, Yong Xiang, Guang Hua, and Song Guo. Protection of big data privacy. *IEEE access*, 4:1821–1834, 2016.

[141] Priyank Jain, Manasi Gyanchandani, and Nilay Khare. Big data privacy: a technological perspective and review. *Journal of Big Data*, 3(1):1–25, 2016.

[142] Shiqing Ma, Kyu Hyung Lee, Chung Hwan Kim, Junghwan Rhee, Xiangyu Zhang, and Dongyan Xu. Accurate, low cost and instrumentation-free security audit logging for windows. In *Proceedings of the 31st Annual Computer Security Applications Conference*, pages 401–410, 2015.

[143] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[144] Vasileios Mavroeidis, Kamer Vishi, and Audun Jøsang. A framework for data-driven physical security and insider threat detection. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1108–1115, 2018.

[145] Shuhan Yuan and Xintao Wu. Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security*, page 102221, 2021.

[146] Solane Duque and Mohd Nizam bin Omar. Using data mining algorithms for developing a model for intrusion detection system (ids). *Procedia Computer Science*, 61:46–51, 2015.

[147] Abdulrahman Alharby and Hideki Imai. Ids false alarm reduction using continuous and discontinuous patterns. In *Proceedings of the International Conference on Applied Cryptography and Network Security*, pages 192–205, 2005.

[148] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

[149] Jeremiah Blocki, Nicolas Christin, Anupam Datta, Ariel Procaccia, and Arunesh Sinha. Audit games with multiple defender resources. In *Proceedings of the 2015 AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[150] Verizon. 2020 data breach investigations report. https://enterprise.verizon.com/resources/reports/dbir/, 2020.

[151] Robert J Aumann. Rationality and bounded rationality. In *Cooperation: Game-Theoretic Approaches*, pages 219–231. Springer, 1997.

[152] James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[153] Rong Yang, Fernando Ordonez, and Milind Tambe. Computing optimal strategy against quantal response in security games. In *Proceedings of the 2012 International Conference on Autonomous Agents and Multi-agent Systems*, pages 847–854, 2012.

[154] Manish Jain, James Pita, Milind Tambe, Fernando Ordónez, Praveen Paruchuri, and Sarit Kraus. Bayesian stackelberg games and their application for security at los angeles international airport. *ACM SIGecom Exchanges*, 7(2):1–3, 2008.

[155] Verizon. Protected health information data breach report. http://www.verizonenterprise.com/resources/protected_health_information_data_breach_report_en_xg.pdf, Feb 2018.

[156] Protenus. 2021 breach barometer. http://https://email.protenus.com/hubfs/2021%20Breach%20Barometer.pdf, Mar 2021.

[157] Bo An, Milind Tambe, Fernando Ordonez, Eric Shieh, and Christopher Kiekintveld. Refinement of strong stackelberg equilibria in security games. In *Proceedings of the 2011 AAAI Conference on Artificial Intelligence*, volume 25, 2011.

[158] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 2008 International Joint Conference on Autonomous Agents and Multi-agent Systems*, pages 895–902, 2008.

[159] John Conlisk. Why bounded rationality? *Journal of economic literature*, 34(2):669–700, 1996.

[160] James Pita, Richard John, Rajiv Maheswaran, Milind Tambe, and Sarit Kraus. A robust approach to addressing human adversaries in security games. In *Proceedings of the European Conference on Artificial Intelligence*, pages 660–665, 2012.

[161] Fouzia F Ozair, Nayer Jamshed, Amit Sharma, and Praveen Aggarwal. Ethical issues in electronic health records: A general overview. *Perspectives in clinical research*, 6(2):73, 2015.

[162] Akhil Shenoy and Jacob M Appel. Safeguarding confidentiality in electronic health records. *Cambridge Quarterly of Healthcare Ethics*, 26(2):337–341, 2017.

[163] Bradley Malin and Edoardo Airoldi. Confidentiality preserving audits of electronic medical record access. *Studies in health technology and informatics*, 129(1):320, 2007.

[164] Ziqi Zhang, Chao Yan, Diego A Mesa, Jimeng Sun, and Bradley A Malin. Ensuring electronic medical record simulation through better training, modeling, and evaluation. *Journal of the American Medical Informatics Association*, 27(1):99–108, 2020.

[165] Chao Yan, Ziqi Zhang, Steve Nyemba, and Bradley A Malin. Generating electronic health records with multiple data types and constraints. In *Proceedings of the 2020 AMIA Annual Symposium Proceedings*, volume 2020, page 1335, 2020.

[166] Ziqi Zhang, Chao Yan, Thomas A Lasko, Jimeng Sun, and Bradley A Malin. Synteg: a framework for temporal structured electronic health data simulation. *Journal of the American Medical Informatics Association*, 28(3):596–604, 2021.