To Warn or Not to Warn: Online Signaling in Audit Games

Chao Yan
Dept. of EECS
Vanderbilt University
Nashville, USA
chao.yan@vanderbilt.edu

Haifeng Xu
Dept. of Computer Science
University of Virginia
Charlottesville, USA
hx4ad@virginia.edu

Yevgeniy Vorobeychik
Dept. of Computer Science and Engineering
Washington University at St. Louis
St. Louis, USA
yvorobeychik@wustl.edu

Bo Li
Dept. of Computer Science
University of Illinois at Urbana-Champaign
Champaign, USA
lbo@illinois.edu

Daniel Fabbri
Dept. of Biomedical Informatics
Vanderbilt University Medical Center
Nashville, USA
daniel.fabbri@vumc.org

Bradley A. Malin
Dept. of Biomedical Informatics
Vanderbilt University Medical Center
Nashville, USA
b.malin@vumc.org

***Abstract*—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 paper, 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.**

***Index Terms*—database auditing, privacy, signaling, Stackelburg game**

I. 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 [1], [2]. 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) [3], [4]. 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 [5]. 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 [6]–[8]. 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 [9], [10]. 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 [11]–[14]. For instance, many healthcare organizations (HCOs) rely on alert, as well auditing, mechanisms to monitor anomalous accesses to electronic medical records (EMRs) by employees who may violate policy and breach the privacy of certain patients [15]. 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 [16]. 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

This work was supported, in part by grant R01LM10207 from the National Institutes of Health, grants CNS-1526014 and IIS-1905558 from the National Science Foundation, and grant W911NF-19-1-0241 from the Army Research Office. Haifeng Xu is supported by a Google Faculty Research Award.

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00048

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 [17]. 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 [18]–[20]. 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 LAPD to randomize checkpoints on the roadways at Los Angeles International Airport) [21] and *IRIS* (which was adopted by the US Federal Air Marshal Service to schedule air marshals on international flights) [22]. The audit game is a variation of the SSG designed to discover an efficient audit strategy [23]–[26]. 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 [27]. This can be addressed by strategically revealing information to the attacker [27], [28], a mechanism referred to as *signaling* (or *persuasion* [29], [30]). 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 EMRs 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.

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

In this paper, 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 paper we evaluate the expected utility of the auditor with a dataset of over 10 million real VUMC EMR 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 paper 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. Finally, we review representative related research in the database auditing domain, with a focus on methodology in the adversarial setting.

II. 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.

A. Motivating Domain

To provide efficient healthcare service, HCOs typically store and process each patient's clinical, demographic, and financial information in an EMR system. EMR users, such as physicians and other clinical staff, need to access patients' EMRs when providing healthcare services. The routine workflow can be summarized as three steps: 1) a user initiates a search for a patient's EMR 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' EMRs [31].

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 EMR of co-workers, neighbors, family members, and VIPs [15]. 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.

B. 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 $\sum_t \theta^t \cdot V^t d^t \le B$.

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 EMR 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 $U_{d,u}^t$ and $U_{a,u}^t$, respectively. On the other hand, if the auditor audits a victim alert of type t, the auditor and the attacker will receive utility $U_{d,c}^t$ and $U_{a,c}^t$, respectively. Here, the subscripts d and a stand for *defender* and *attacker*, respectively. Naturally, we assume $U_{a,c}^t < 0 < U_{a,u}^t$ and $U_{d,c}^t \ge 0 > U_{d,u}^t$.

Figure 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.

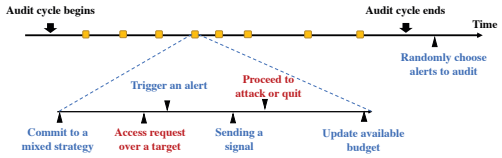


Fig. 1. The auditor and attacker actions are shown in *blue* and *red*, respectively.

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

$$\begin{aligned} &\mathbf{P}(\xi_1^\tau, X_c^\tau) = p_1^\tau, \quad &&\mathbf{P}(\xi_1^\tau, X_u^\tau) = q_1^\tau, \\ &\mathbf{P}(\xi_0^\tau, X_c^\tau) = p_0^\tau, \quad &&\mathbf{P}(\xi_0^\tau, X_u^\tau) = q_0^\tau. \end{aligned} \tag{1}$$

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 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,

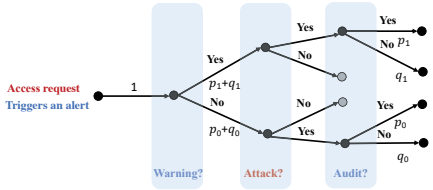


Fig. 2. The decision tree of the auditor and an arbitrary user, the actions for which are shown in *blue* and *red*, respectively.

observe that, $p_1 + q_1 + p_0 + q_0 = 1$, and the overall probability of auditing this alert is $\mathbf{P}(X_c) = \mathbf{P}(X_c, \xi_1) + \mathbf{P}(X_c, \xi_0) = p_1 + p_0$.

[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.

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:

- 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 [27], 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).

III. 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.

A. 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 $\boldsymbol{D}_\tau^t$, 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 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 $\boldsymbol{B}_\tau = \{B_\tau^1, B_\tau^2, ..., B_\tau^{|T|}\}$ 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:

$$
\begin{aligned}
& \max_{\boldsymbol{B}_\tau} \theta^t(t)\cdot U_{d,c}^t + (1-\theta^t(t))\cdot U_{d,u}^t \\
s.t.\quad & \\
& \forall t', \quad \theta^t(t)\cdot U_{a,c}^t + (1-\theta^t(t))\cdot U_{a,u}^t \\
& \qquad\qquad \geq \theta^{t'}(t)\cdot U_{a,c}^{t'} + (1-\theta^{t'}(t))\cdot U_{a,u}^{t'}, \\
& \forall t', \quad \theta^{t'}(t) = \mathbb{E}_{d_\tau^{t'}\sim \boldsymbol{D}_\tau^{t'}}\left(\frac{B_\tau^{t'}}{V^{t'} d_\tau^{t'}}\right), \\
& \qquad \sum_{t'=1}^{|T|} B_\tau^{t'} \leq B_\tau, \\
& \forall t', \quad B_\tau^{t'} \in [0, B_\tau],
\end{aligned}
\tag{2}
$$

where the first constraint ensures that t is the attacker's best response. After solving $|T|$ instances of LP (2), the best solution for the auditor will henceforth be referred to as the *online SSE strategy* (or simply, the *SSE*), $\boldsymbol{\theta}_{SSE}$.

B. 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

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

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:

$$\begin{aligned}
\max_{\boldsymbol{p_0},\boldsymbol{p_1},\boldsymbol{q_0},\boldsymbol{q_1},\boldsymbol{B}_\tau} \quad & p_0^t \cdot U_{d,c}^t + q_0^t \cdot U_{d,u}^t + \sum_{t'=1}^{|T|}(p_1^{t'} + q_1^{t'}) \cdot P^{t'} \cdot E_\tau^{t'} \cdot C_{t'} \\
s.t. \quad & \\
& \forall t', \quad p_0^t \cdot U_{a,c}^t + q_0^t \cdot U_{a,u}^t \geq p_0^{t'} \cdot U_{a,c}^{t'} + q_0^{t'} \cdot U_{a,u}^{t'}, \\
& \forall t', \quad p_1^{t'} \cdot U_{a,c}^{t'} + q_1^{t'} \cdot U_{a,u}^{t'} \leq 0, \\
& \forall t', \quad p_1^{t'} + p_0^{t'} = \mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}}\left(\frac{B_\tau^{t'}}{V^{t'} d_\tau^{t'}}\right), \\
& \forall t', \quad p_1^{t'} + p_0^{t'} + q_1^{t'} + q_0^{t'} = 1, \\
& \quad \sum_{t' \in \{1,\ldots,|T|\}} B_\tau^{t'} \leq B_\tau, \\
& \forall t', \quad B_\tau^{t'} \in [0, B_\tau], \\
& \forall t', \quad p_0^{t'}, q_0^{t'}, p_1^{t'}, q_1^{t'} \in [0,1],
\end{aligned} \tag{3}$$

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 (3) we use the expected number of future alerts $\mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}}(d_\tau^{t'})$ to approximate $\boldsymbol{E}_\tau^{t'}$. As a result, $\mathbb{E}_{d_\tau^{t'} \sim \boldsymbol{D}_\tau^{t'}}(d_\tau^{t'})$ 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 $\boldsymbol{p_0}$, $\boldsymbol{p_1}$, $\boldsymbol{q_0}$ and $\boldsymbol{q_1}$ 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 (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 (3) as the *Online Stackelberg Signaling Policy (OSSP)*. In particular, we use $\boldsymbol{\theta}_{ossp}$ 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.

C. 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 $\mathbb{E}_{d_\tau^t \sim \boldsymbol{D}_\tau^t}(d_\tau^t)$, 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 $\mathbb{E}_{d_{\tau-1}^t \sim \boldsymbol{D}_{\tau-1}^t}(d_{\tau-1}^t)$ 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.

IV. 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 1. *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 (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 (3) and the two equations are by our definition of the new variables. This proves that the first constraint is also feasible.

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 1 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 2. *Let θ_{ossp}^t be the marginal coverage probability in the OSSP at any given game status and θ_{SSE}^t be the corresponding marginal coverage probability in the online SSE. Then, in a SAG, for each type $t \in T$, $\theta_{ossp}^t = \theta_{SSE}^t$.*

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, $\theta_{ossp}^t = \theta_{SSE}^t$ 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, $\forall t \neq t_*, \mathbb{E}_a(\theta_{ossp}^t) = \mathbb{E}_a^{ossp}$. 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 (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 3. *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 (3), which, in this case, share exactly the same form with LP (2). In combination with Theorem 1, 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: $\mathbb{E}_d(\theta^t) = \theta^t \cdot U_{d,c}^t + (1 - \theta^t) \cdot U_{d,u}^t$. 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.

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

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., $p_0^{t_*} = 0$).

Theorem 4. *In SAG, if the payoff structure satisfies* $0 \geq (U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) / (U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) \geq U_{a,c}^{t_*} / U_{a,u}^{t_*}$ *on the best attacking type* t_* *in the OSSP, then we have* $p_0^{t_*} = 0$ *on the* τ*–th alert.*

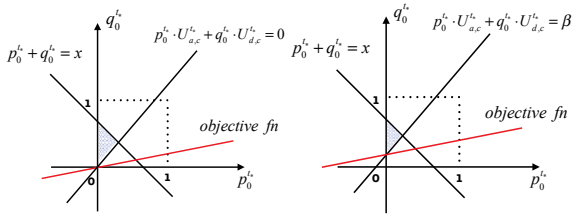
Proof. This will be proved in the instance of LP (3) that derives the best pair of the signaling strategy and the attacking strategy t_*. For inference convenience, for all t we substitute p_1^t and q_1^t with $\theta_{ossp}^t - p_0^t$ and $1 - \theta_{ossp}^t - q_0^t$, respectively. Combining with Theorem 1, the objective function of LP (3) can be simplified as $p_0^{t_*} \cdot U_{d,c}^{t_*} + q_0^{t_*} \cdot U_{d,u}^{t_*} + p_1^{t_*} \cdot P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*} + q_1^{t_*} \cdot P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*} = p_0^{t_*} \cdot (U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) + q_0^{t_*} \cdot (U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) + P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$.

Now, we simplify constraints. The first constraint is always tight in the OSSP (as shown in Theorem 2). By applying the substitution rules, the second constraint becomes $\forall t'$, $p_0^{t'} \cdot U_{a,c}^{t'} + q_0^{t'} \cdot U_{a,u}^{t'} \geq \theta_{ossp}^{t'} \cdot U_{a,c}^{t'} + (1 - \theta_{ossp}^{t'}) \cdot U_{a,u}^{t'}$. For all $t' \neq t_*$, it can be future transformed into $(\theta_{ossp}^{t'} - p_0^{t'}) \cdot U_{a,c}^{t'} + (1 - \theta_{ossp}^{t'} - q_0^{t'}) \cdot U_{a,u}^{t'} \leq 0$. Due to the fact that $p_1^{t'} = q_1^{t'} = 0$ in the OSSP for $\forall t' \neq t_*$, $p_0^{t'}$ is equal to $\theta_{ossp}^{t'}$, and $q_0^{t'}$ equal to $1 - \theta_{ossp}^{t'}$. 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 $p_0^{t_*} \cdot (U_{d,c}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) + q_0^{t_*} \cdot (U_{d,u}^{t_*} - P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}) + P^{t_*} \cdot E_\tau^{t_*} \cdot C_{t_*}$, such that $p_0^{t_*} \cdot U_{a,c}^{t_*} + q_0^{t_*} \cdot U_{a,u}^{t_*} \geq \theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*}$ (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 $\mathbb{E}_\alpha^{t_*}(util|\xi_u) = p_0^{t_*} \cdot U_{a,c}^{t_*} + q_0^{t_*} \cdot U_{a,u}^{t_*} > 0$, 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 3(a). 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 3(b). The OSSP is $p_1^{t_*} = \theta_{ossp}^{t_*}, p_0^{t_*} = 0, q_1^{t_*} = 1 - \theta_{ossp}^{t_*} - (\theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*}) / U_{a,u}^{t_*}, q_0^{t_*} = (\theta_{ossp}^{t_*} \cdot U_{a,c}^{t_*} + (1 - \theta_{ossp}^{t_*}) \cdot U_{a,u}^{t_*}) / U_{a,u}^{t_*}$. □

[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) $\beta > 0$

Fig. 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.

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 the number of future warning events, which decreases with time, the condition in Theorem 4 only happens in a certain period of time.

One might wonder that, given that the condition in Theorem 4 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.

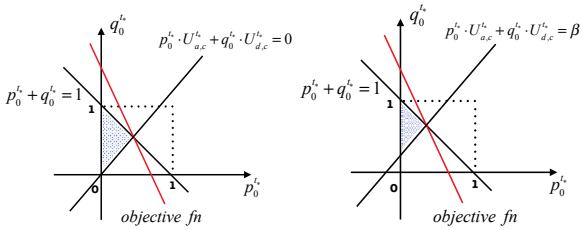
A natural follow-up question is can the attacker manipulate the model by running this strategy across audit cycles? The answer is no as well. Such a behavior can be easily detected by a rule that applies when the attacker performs his/her attack repeatedly. When the auditor does not send a warning, the

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 5. *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 4. 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(a) and 4(b), 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 1 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 (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 (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) $\beta > 0$

Fig. 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.

TABLE I
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 (≤ 0.5 miles)	140.46	23.23
4	Same Address	10.84	3.73
5	Last Name; Neighbor (≤ 0.5 miles)	25.43	4.51
6	Last Name; Same Address	15.14	4.10
7	Last Name; Same Address; Neighbor (≤ 0.5 miles)	43.27	6.45

V. Model Evaluation

In this section, we evaluate the performance of the SAG on the real EMR 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 II
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

A. Dataset

The dataset consists of EMR 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 ⟨Date, Employee, Patient⟩ 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 I 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 II. These values are estimates based on discussions with experts working in the area.

B. 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 (3)) and both the *offline* and *online SSE* (the optimal objective value of LP (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 4), 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 $U^1_{a,c} = \{-500, -1000, -1500, -2000, -2500, -3000, -3500\}$ and fixing the other variables to their values in Table II. We set $U^1_{d,c} = \{25, 50, 75, 100, 125, 150, 175\}$ and run the same evaluation as described above.

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

C. Results

We considered all 7 alert types described in Table I. Due to space limitations, we only show the sequential results of 15 sequential testing days along the timeline in Figures 5(a)-5(o) by applying $B = 50, C_t = -1$ for all types and $\alpha = 1\%$.

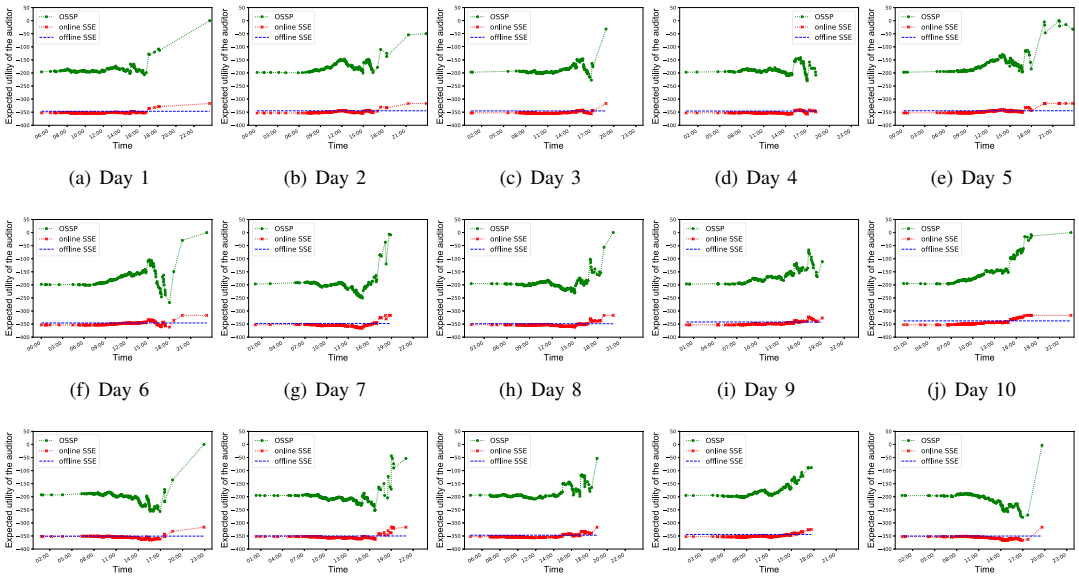
It is noteworthy that the type for each alert may not be aligned with the optimal attacking type in the OSSP strategy. Thus, to compare the approaches, we only apply the SAG on alerts whose type is equal to the best attacking type in the OSSP. For alerts whose types differ, we simply apply the online 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(a)-5(o) 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 5(a), 5(f), 5(g), 5(h), 5(j), 5(k) and 5(o), 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

[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.



(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

Fig. 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 III

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 IV

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.

$U^1_{a,c}$	-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
$U^1_{d,c}$	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

the budget for inspecting anomalous repeated requests. The results are shown in Table III, 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 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 IV, 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.

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,

TABLE V
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

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 V 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.

VI. RELATED WORK

There have been a number of investigations into effective alert management strategies and efficient auditing mechanisms for database systems. In this section, we review the game-theoretic developments that are related to our investigation.

Blocki *et al.* first modeled the audit problem between an auditor and an auditee as a classic security game. 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 [23]. To simulate the real audit environment, Blocki *et al.* generalized the framework by accounting for the situations with multiple defender resources [24]. 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 database audit setting. To solve this challenge, Yan *et al.* introduced a game theoretic audit approach to 1) prioritize the order in which types of alerts are investigated and 2) provide an upper bound on how much resource to allocate for auditing each type [25], [26], [32]. Schlenker *et al.* introduced an approach dealing with how to assign alerts to security analysts was proposed, where each analyst has different areas of expertise [33]. However, all of these investigations adopted a classic security game framework, which, as our experiments show, hinder the efficacy of the system.

It has been shown that the integration of a signaling mechanism into adversarial settings can improve protection. 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 [27]. The advantages of signaling were subsequently extended to Bayesian Stackelberg games, where players have payoff-relevant private information [30]. 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 [34]. 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.

VII. DISCUSSION

In this paper, 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 EMR 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 paper 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 paper 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 [35]. Third, in this paper, 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.

VIII. 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.

References

[1] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012.

[2] S. Yin and O. Kaynak, "Big data for modern industry: challenges and trends [point of view]," *Proceedings of the IEEE*, vol. 103, no. 2, pp. 143–146, 2015.

[3] K. E. Pavlou and R. T. Snodgrass, "Dragoon: An information accountability system for high-performance databases," in *28th IEEE International Conference on Data Engineering*, 2012, pp. 1329–1332.

[4] D. Fabbri, R. Ramamurthy, and R. Kaushik, "Select triggers for data auditing," in *Proceedings of the 29th IEEE International Conference on Data Engineering*, 2013, pp. 1141–1152.

[5] S. A. 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.

[6] D. S. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in big data," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions*, 2015, pp. 202–207.

[7] R. Hasan, S. Zawoad, S. Noor, M. M. Haque, and D. Burke, "How secure is the healthcare network from insider attacks? an audit guideline for vulnerability analysis," in *Proceedings of the IEEE 40th Annual Computer Software and Applications Conference*, vol. 1, 2016, pp. 417–422.

[8] M. Puppala, T. He, X. Yu, S. Chen, R. Ogunti, and S. T. 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*, 2016, pp. 5–8.

[9] R. Motwani, S. U. Nabar, and D. Thomas, "Auditing sql queries," in *Proceedings of the 24th IEEE International Conference on Data Engineering*, 2008, pp. 287–296.

[10] H. Mazzawi, G. Dalal, D. Rozenblatz, L. Ein-Dorx, M. Niniox, and O. Lavi, "Anomaly detection in large databases using behavioral patterning," in *Proceedings of the 33rd IEEE International Conference on Data Engineering*, 2017, pp. 1140–1149.

[11] H. D. Kuna, R. García-Martinez, and F. R. Villatoro, "Outlier detection in audit logs for application systems," *Information Systems*, vol. 44, pp. 22–33, 2014.

[12] J. Blocki, N. Christin, A. Datta, and A. 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*, 2012, pp. 38–59.

[13] W. Lu and G. Miklau, "Auditing a database under retention restrictions," in *Proceedings of the 25th IEEE International Conference on Data Engineering*, 2009, pp. 42–53.

[14] S. M. Groomer and U. S. Murthy, "Continuous auditing of database applications: An embedded audit module approach," in *Continuous Auditing: Theory and Application*, 2018, pp. 105–124.

[15] M. Hedda, B. Malin, C. Yan, and D. Fabbri, "Evaluating the effectiveness of auditing rules for electronic health record systems," in *Proceedings of the 2017 AMIA Annual Symposium*, vol. 2017, 2017, pp. 866–875.

[16] A. Barth, J. Mitchell, A. Datta, and S. Sundaram, "Privacy and utility in business processes," in *Proceedings of the 20th IEEE Computer Security Foundations Symposium*, 2007, pp. 279–294.

[17] A. Laszka, Y. Vorobeychik, D. Fabbri, C. Yan, and B. Malin, "A game-theoretic approach for alert prioritization," in *Proceedings of the 31st AAAI Workshop on Artificial Intelligence for Cyber Security*, 2017.

[18] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, M. Tambe, A. 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.

[19] C. T. Do, N. H. Tran, C. Hong, C. A. Kamhoua, K. A. Kwiat, E. Blasch, S. Ren, N. Pissinou, and S. S. Iyengar, "Game theory for cyber security and privacy," *ACM Computing Surveys*, vol. 50, no. 2, p. 30, 2017.

[20] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe, "Stackelberg security games: looking beyond a decade of success," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 5494–5501.

[21] J. Pita, M. Jain, F. Ordónez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Using game theory for los angeles airport security," *AI magazine*, vol. 30, no. 1, pp. 43–43, 2009.

[22] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordónez, "Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service," *Interfaces*, vol. 40, no. 4, pp. 267–290, 2010.

[23] J. Blocki, N. Christin, A. Datta, A. D. Procaccia, and A. Sinha, "Audit games." in *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, 2013, pp. 41–47.

[24] J. Blocki, N. Christin, A. Datta, A. D. Procaccia, and A. Sinha, "Audit games with multiple defender resources." in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, vol. 15, 2015, pp. 791–797.

[25] C. Yan, B. Li, Y. Vorobeychik, A. Laszka, D. Fabbri, and B. Malin, "Get your workload in order: Game theoretic prioritization of database auditing," in *Proceedings of the 34th IEEE International Conference on Data Engineering*, 2018, pp. 1304–1307.

[26] C. Yan, B. Li, Y. Vorobeychik, A. Laszka, D. Fabbri, and B. Malin, "Database audit workload prioritization via game theory," *ACM Transactions on Privacy and Security*, vol. 22, no. 3, p. 17, 2019.

[27] H. Xu, Z. Rabinovich, S. Dughmi, and M. Tambe, "Exploring information asymmetry in two-stage security games." in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015, pp. 1057–1063.

[28] Z. Rabinovich, A. X. Jiang, M. Jain, and H. Xu, "Information disclosure as a means to security," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 645–653.

[29] E. Kamenica and M. Gentzkow, "Bayesian persuasion," *American Economic Review*, vol. 101, no. 6, pp. 2590–2615, 2011.

[30] S. Dughmi and H. Xu, "Algorithmic bayesian persuasion," in *Proceedings of the 48th annual ACM symposium on Theory of Computing*, 2016, pp. 412–425.

[31] C. A. Gunter, D. Liebovitz, and B. Malin, "Experience-based access management: A life-cycle framework for identity and access management systems," *IEEE Security & Privacy*, vol. 9, no. 5, p. 48, 2011.

[32] L. Tong, A. Laszka, C. Yan, N. Zhang, and Y. Vorobeychik, "Finding needles in a moving haystack: Prioritizing alerts with adversarial reinforcement learning," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[33] A. Schlenker, H. Xu, M. Guirguis, C. Kiekintveld, A. Sinha, M. Tambe, S. Sonya, D. Balderas, and N. 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*, 2017, pp. 381–387.

[34] H. Xu, K. Wang, P. Vayanos, and M. 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.

[35] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proceedings of the 7th ACM conference on Electronic commerce*, 2006, pp. 82–90.