On the Tradeoff between Privacy and Utility in Collaborative Security—A Game Theoretical Approach

Richeng Jin*, Xiaofan He‡, Peng Ning†, Rudra Dutta‡, Huaiyu Dai*,
*Department of ECE, North Carolina State University
Emails: {rjin2,hdai}@ncsu.edu
‡Department of CSC, North Carolina State University
Emails: {pning.dutta}@ncsu.edu
†Department of EE, Lamar University
Email: xhe1@lamar.edu

Abstract—With the rapid development of sophisticated attack techniques, individual security systems which base all of their decisions and actions of attack prevention and response on their own observations and knowledge become incompetent. To cope with this problem, collaborative security that coordinates security entities to perform specific security actions is proposed and developed in literature. In collaborative security schemes, multiple entities collaborate with each other by sharing some security evidence or analysis results so as to make more effective and reasonable decisions. Nevertheless, the information exchange raises privacy concerns, especially for those privacy-sensitive entities. In order to obtain a quantitative understanding of the fundamental tradeoff between the effectiveness of collaboration and the entities’ privacy, a repeated two-layer single-leader multi-follower game is proposed in this work. Based on our game-theoretic analysis, the expected behaviors of both the attacker and the security entities are derived and the utility-privacy tradeoff curve is obtained. In addition, the existence of Nash equilibrium (NE) is proved and an asynchronous dynamic update algorithm is proposed to compute the optimal collaboration strategies of the entities. Furthermore, the existence of Byzantine entities is considered and its influence is investigated. Finally, simulation results are shown to validate the analysis.

I. INTRODUCTION

Individual security systems are supposed to base all of their decisions and actions to prevent and respond to attacks on their own observations and (often limited) knowledge. With the development of sophisticated large-scale attack techniques, it becomes more and more difficult for individual security systems to provide effective security service. To mitigate this problem, collaborative security is proposed and developed [1].

Collaborative security has been widely applied and proven to be an effective approach in many security domains including intrusion detection, anti-spam, anti-malware, identification of insider attackers and detection of botnet (see, e.g., [2, 3] and the references therein). The objective of the entities in the collaborative security schemes is to collaborate and contribute their efforts by sharing security-related information with each other. In such collaborative environments, considering that some confidential information may be leaked in the information sharing procedure, some techniques have been proposed to protect the privacy [4–12], at the cost of utility loss (i.e., a collaboration effectiveness degradation). However, there are two major limitations in these pioneering works. Firstly, it is often difficult to quantify the amount of preserved privacy and utility loss in the existing methods. Secondly, the existing methods do not have the flexibility of properly adjusting the collaboration strategies in response to a given privacy requirement.

In this work, a new privacy-aware collaboration scheme is proposed for collaborative security, which is amenable to the quantitative utility-privacy tradeoff analysis and flexible in meeting the pre-specified privacy requirement. Considering the self-interestedness of the entities and the intelligence of the attacker, a game-theoretic approach is taken in this work. More specifically, the interaction among the attacker and the group of collaborative security entities is modeled as a two-layer game. The first-layer focuses on the interaction between the attacker and the entities. Particularly, the influence of the privacy requirement on the entities’ responding strategies and the overall detection performance is explored, and based on which, the corresponding utility-privacy tradeoff curve is obtained. The second-layer focuses on the interactions among security entities themselves and based on which, the optimal collaboration strategies of the entities in different scenarios are derived. In addition, the existence of Byzantine entities is further considered and its influence is investigated.

The remainder of this paper is organized as follows. Section II formulates the utility-privacy tradeoff problem. The proposed two-layer game model is presented in Section III. The proposed game is solved in Section IV. The existence of Byzantine entities is discussed in Section V. The theoretical analysis is validated through simulations in Section VI. Related works are discussed in Section VII. Conclusions and future works are presented in Section VIII.

II. PROBLEM FORMULATION

In this work, a network that consists of $N$ different self-interested security entities is considered, denoted by $N = \{1, 2, \ldots, N\}$. Let $s_t$ denote the state of the network at time $t$. 

...
A. Attacker Model

A smart attacker that can infer the possible responding strategies and collaboration strategies of the security entities and choose its optimal attacking strategy accordingly is considered. It is assumed that the attacker is able to manipulate the state of the network by launching attacks and the objective of the attacker is to attack the network without being detected.

Furthermore, for the ease of presentation, it is assumed that the attacker will only launch one type of attack (e.g., DDoS) on the network.\(^1\) As a result, the network has two possible states, i.e., \(s_t \in \{0, 1\}\) in which \(s_t = 1\) (\(s_t = 0\)) stands for abnormal (normal) state corresponding to the case that the attacker launches (does not launch) an attack. If the attacks are responded by the security entities successfully, the attacker will not attack the network again (in the time frame of interest) using the same type of attack, and thus lose the control of the network state. This assumption makes sense because an attacker usually launches an attack by exploiting the vulnerabilities of the system and once the attack is detected and identified, the vulnerabilities will be fixed and relevant signatures be recorded, which makes the same attack ineffective. If the attacker switches to a new type of attack, it is equivalent to start a new game in our model, which is hence not considered here for simplicity.

The action space of the attacker against the network is \(A = \{a_1, a_2\}\), where \(a_1\) corresponds to “attack” and \(a_2\) corresponds to “no attack”. The mixed strategy chosen by the attacker at time \(t\) is denoted by \(p_t^A = [p_t^A(a_1), p_t^A(a_2)]\), in which \(p_t^A(a_1)\) and \(p_t^A(a_2)\) are the probabilities that the attacker takes action \(a_1\) and \(a_2\) at time \(t\), respectively.

B. Defender Model

At time \(t\), each security entity \(j\) in the network will independently obtain a private observation (denoted by \(Y_{j,t}\)) about the network state \(s_t\), which is unknown to the security entities. Each entity \(j\) knows the structure of its private observation, which is represented by a set of parameterized marginal distributions \(Q^j = \{q_j(Y_{j,t} | s_t) | Y_{j,t} \in \{0, 1\}\}\), where \(q_j(\cdot | s_t)\) is the distribution of the private observation given the true network state \(s_t\).

Example 1: Each security entity \(j\) may deploy Intrusion Detection Systems (IDSs) to monitor the network traffics for intrusion detection. When the attacker launches an attack, the network state becomes abnormal (i.e., \(s_t = 1\)), and the IDSs output the detection result \(Y_{j,t}\). In this case, the parameterized marginal distributions depend on the detecting capability of the IDSs. More specifically, \(q_j(Y_{j,t} = 1 | s_t = 1)\) and \(q_j(Y_{j,t} = 1 | s_t = 0)\) correspond to the detection rate and false alarm rate of the IDSs, respectively.

Since the private observations may not be sufficient for the entities to learn the true network state \(s_t\) individually, this work considers the scenario in which the security entities in the network can collaborate and share their private observations so as to further enhance the network security. However, sharing the private observations may lead to potential privacy leakage for the entities. For instance, if entity \(j\) observes the network state correctly (e.g., detects an attack on the network successfully), it indicates that entity \(j\) has the corresponding detection knowledge (e.g., signatures). By knowing the observations, the attacker can infer the security state (e.g., whether the entity knows the signatures of the attack) of the corresponding entity and therefore choose a better attacking strategy. Moreover, continuing Example 1, an intrusion alert usually contains some private information (e.g., IP address, processing time), which may raise big privacy concerns for the entities. As a result, the security entities should also balance their utilities and privacy concerns so as to choose proper collaboration strategies. With such consideration, in order for privacy preservation, each entity \(j\) shares an obfuscated version of \(Y_{j,t}\) with others, denoted by \(\hat{Y}_{j,t}\). In this work, it is assumed that each entity \(j\) will misreport its true observation result to other entities with probability \(p_j^0\) and the preserved privacy is measured by the entropy introduced by \(p_j^0\) [13], given as follows:

\[H(p_j^0) = -p_j^0 \log_2(p_j^0) - (1 - p_j^0) \log_2(1 - p_j^0).\]  \(1\)

Among all the security entities, it is further assumed that there exists a master node (denoted by entity \(i\)) in the network which will collect the shared observations and suggest a recommended action for the other entities to follow. To this end, the objective of entity \(i\) is to respond to the attacks properly on behalf of all the entities when the network is under attack. Its action space is \(D = \{d_1, d_2\}\), where \(d_1\) corresponds to “respond” and \(d_2\) corresponds to “do nothing”. The mixed strategy chosen by entity \(i\) is denoted by \(p_i^D = [p_i^D(d_1), p_i^D(d_2)]\), in which \(p_i^D(d_1)\) and \(p_i^D(d_2)\) are the probabilities that it takes action \(d_1\) and \(d_2\) at time \(t\), respectively. Note that since the other entities will follow the recommended action of entity \(i\), the action and payoff of entity \(i\) are used to represent those of the whole network in the following discussion.

C. General Settings

Let \(W\) denote the loss of security when the attacker launches an attack without being detected. If entity \(i\) fails to respond to the attacker successfully, the attacker gets a payoff \(-W\) and all the entities in the network get a payoff \(-W\). Otherwise, the payoffs for the attacker and the entities are \(-W\) and \(-W\), respectively. Table I illustrates the payoff matrix of the attacker/entities interaction, in which the first entry and second entry in each cell denote the payoffs of the attacker and the entities, respectively. In the matrix, \(b \in [0, 1]\) denotes the possibility of successful response for the entities; similar to \([14]\), the cost of attacking and responding are assumed to be proportional to \(W\), denoted by \(C_a W\) and \(C_r W\), respectively, in which \(C_a\) and \(C_r\) denote the corresponding cost coefficients. When the attacker chooses to attack and the entities choose to respond at the same time, the probability of successful response for the network is \(b\), which means all the entities

\(^1\)When multiple types of attacks are available to the attacker, multiple independent games can be formed, each corresponding to a different type of attack.
will get payoff $W - C_r W$ and $-W - C_r W$ with probability $b$ and $1 - b$, respectively. Therefore, the payoff of the entities in expectation is $-(1-2b)W - C_r W$. Similarly, the payoff of the attacker is $(1-2b)W - C_a W$. The payoffs of both the attacker and the entities in other cases are defined similarly.

Note that when the entities choose “do nothing”, the payoff of the attacker choosing “attack” should be higher than that of choosing “no attack” (otherwise, the attacker has no incentive to attack), which indicates $C_a < 1$. Similarly, $C_r < 1$.

### III. Collaborative Security Game Model

In this section, the problem is modeled as a repeated two-layer single-leader multi-follower game, in which the attacker acts as the leader and entities act as the followers. The first-layer game models the interaction between the attacker and entity $i$, while the second-layer game models the collaborative information sharing among the entities themselves. Figure 1 depicts a special case of the game model in which there are only three collaborative entities. More specifically, the problem is solved in two steps: first of all, the first-layer game between the attacker and entity $i$ is solved, which determines the optimal payoffs of both the attacker and entity $i$ as functions of the collaboration strategies of the entities. Then, based on the payoff functions from the first-layer game, the entities further determine their optimal collaboration strategies given their privacy requirements in the second-layer game.

#### A. The First-layer Leader-follower Game

In the first-layer game, it is assumed that the follower plays a myopic best-response strategy to the leader’s strategy at each time $t$ [15]. Note that since it is not possible for the entities to know the future strategies of the attacker and the future observations of themselves, the myopic strategy is actually the best strategy that an entity can take.

1) **The Followers’ Problem:** At time $t$, let $\tilde{Y}_{-i,t}$ denote the set of obfuscated observations shared by other entities, given the attacker’s strategy $p_t^A$ and its own observation result $Y_{i,t}$, entity $i$ first estimates the probability that the attacker actually launches an attack, which is given by

$$F_t^i(a_1|Y_{i,t}, \tilde{Y}_{-i,t}) = \frac{p_t^A(a_1)p(Y_{i,t}, \tilde{Y}_{-i,t}|a_1)}{p(Y_{i,t}, \tilde{Y}_{-i,t})},$$

(2)

where $p(Y_{i,t}, \tilde{Y}_{-i,t}|a_1)$ is the probability that the observation of entity $i$ is $Y_{i,t}$ while the shared obfuscated observations are $\tilde{Y}_{-i,t}$ at time $t$ given that the attacker launches an attack; $p(Y_{i,t}, \tilde{Y}_{-i,t})$ is the probability that the observation of entity $i$ is $Y_{i,t}$ while the shared obfuscated observations are $\tilde{Y}_{-i,t}$ at time $t$. They are given by

$$p(Y_{i,t}, \tilde{Y}_{-i,t}|a_1) = q_i(Y_{i,t}|s_t = 1) \prod_{j \neq i} q_j(Y_{j,t}|s_t = 1),$$

(3)

$$p(Y_{i,t}, \tilde{Y}_{-i,t}) = p_t^A(a_1)p(Y_{i,t}, \tilde{Y}_{-i,t}|a_1) + p_t^A(a_2)p(Y_{i,t}, \tilde{Y}_{-i,t}|a_2),$$

(4)

in which $q_i(Y_{i,t}|s_t = 1)$ is the probability that entity $i$ observes $Y_{i,t}$ when $s_t = 1$ and $q_j(Y_{j,t}|s_t = 1)$ is the probability that entity $j$ shares $\tilde{Y}_{j,t}$ with entity $i$ when $s_t = 1$. Then, entity $i$ finds its optimal strategy by solving the following optimization problem:

$$p_t^A(Y_{i,t}, \tilde{Y}_{-i,t}) = \arg \max_{p_t^A} U_t^D(p_t^D, p_t^A, Y_{i,t}, \tilde{Y}_{-i,t}).$$

(5)

The payoff function $U_t^D(p_t^D, p_t^A, Y_{i,t}, \tilde{Y}_{-i,t})$ is given by

$$U_t^D(p_t^D, p_t^A, Y_{i,t}, \tilde{Y}_{-i,t}) = F_t^i(a_1|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_1)[(1-2b)W - C_r W]$$

$$- F_t^i(a_1|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_2)W - F_t^i(a_2|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_1)C_r W,$$

(6)

where $F_t^i(a_1|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_1)$ is the probability of the case that the attacker launches an attack and entity $i$ chooses to respond given the observations $Y_{i,t}, \tilde{Y}_{-i,t}$, and $(1-2b)W - C_r W$ is the payoff of entity $i$ in this case; $F_t^i(a_1|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_2)$ is the probability of the case that the attacker launches an attack and entity $i$ chooses not to do anything given the observations $Y_{i,t}, \tilde{Y}_{-i,t}$, and $-W$ is the payoff of entity $i$ in this case; $F_t^i(a_2|Y_{i,t}, \tilde{Y}_{-i,t})p_t^D(d_1)$ is the probability of the case that the attacker does not launch an attack and entity $i$ chooses to respond given the observations $Y_{i,t}, \tilde{Y}_{-i,t}$, and $-C_r W$ is the payoff of entity $i$ in this case.

2) **The Leader’s Problem:** As the attacker knows that the followers will choose their strategies to maximize their corresponding payoffs, it will choose the strategy that maximizes its own payoff. However, since the attacker does not know the actual observations of the entities, it has to maximize the expected payoff corresponding to the distribution $p(Y_{i,t}, \tilde{Y}_{-i,t})$ which can be obtained by (4) given its chosen strategy. As a result, the attacker finds its optimal strategy by solving the following optimization problem:

$$p_t^D(p_t^A) = \arg \max_{p_t^A} \sum_{t=1}^{T_e} U_t^A(p_t^A, p_t^D(p_t^A)),$$

(7)
where $T_i$ is the time when entity $i$ successfully responds to the attacker and $U^A_t(p^A_t, p^D_t(p^A_t))$ is given by
\[
U^A_t(p^A_t, p^D_t(p^A_t)) = \sum_{Y_{i,t}, \hat{Y}_{i,t} \in \{0,1\}^N} [p(Y_{i,t}, \hat{Y}_{i,t} + t)p^A_t(a_1)p^D_t(d_2|Y_{i,t}, \hat{Y}_{i,t}) (W - C_\alpha W) + p(Y_{i,t}, \hat{Y}_{i,t} - t)p^A_t(a_1)p^D_t(d_1|Y_{i,t}, \hat{Y}_{i,t} - t) [(1 - 2b - C_\alpha) W]],
\] where $p^A_t(a_1)p^D_t(d_2|Y_{i,t}, \hat{Y}_{i,t})$ is the probability of the case that the attacker launches an attack and entity $i$ chooses to do nothing given the observations $Y_{i,t}, \hat{Y}_{i,t}$ and $W - C_\alpha W$ is the payoff of the attacker in this case; $p^A_t(a_1)p^D_t(d_1|Y_{i,t}, \hat{Y}_{i,t} - t)$ is the probability of the case that the attacker launches an attack and entity $i$ chooses to respond given the observations $Y_{i,t}, \hat{Y}_{i,t} - t$ and $(1 - 2b - C_\alpha) W$ is the payoff of the attacker in this case.

B. The Second-layer Game

The second-layer game models the interaction among the entities themselves. In this game, an action of each entity $j$ is a probability $p^c_{j,t} \in [c_1,0.5]^2$ with which the entity $j$ would send out wrong observation result in order to protect its own privacy, and $c_1$ depends on the privacy policy of each entity $j$. The utility function of each entity $j$ is given as follows:
\[
U^D_{j,t}^2(p^c_{j,t}) = R^c_{j,t}(p^c_{j,t}) - R^c_{j,t}(p^c_{j,t} - p^c_{j,t}(s) = 0.5) - \lambda_j P_L(p^c_{j,t}),
\] where $p^c_{j,t} = (p^c_{j,t}(s_1), p^c_{j,t}(s_2), \cdots, p^c_{j,t}(s_{c_1}))$ is a vector which denotes the misreport probabilities of all the entities; $p^c_{j,t}(s) = 0.5$ denotes the misreport probabilities of all the entities other than entity $j$; $R^c_{j,t}(p^c_{j,t})$ denotes the estimated payoff of entity $i$ given $p^c_{j,t}$, which will be discussed in Section IV; $R^c_{j,t}(p^c_{j,t} - p^c_{j,t}(s) = 0.5)$ denotes the estimated reward of entity $i$ when entity $j$ randomly reports its detection result (i.e., $p^c_{j,t}(s) = 0.5$), and therefore $R^c_{j,t}(p^c_{j,t}) - R^c_{j,t}(p^c_{j,t} + p^c_{j,t}(s) = 0.5)$ measures entity $i$'s estimated payoff improvement due to the shared observations from entity $j$; $\lambda_j$ is a constant that measures the importance of privacy loss. The privacy loss $P_L(p^c_{j,t})$ is given by
\[
P_L(p^c_{j,t}) = 1 - H(p^c_{j,t}),
\] where $H(p^c_{j,t})$ denotes the entropy introduce by $p^c_{j,t}$. As a result, each entity $j$ has to solve the following optimization problem:
\[
\begin{align*}
\max_{p^c_{j,t}} U^D_{j,t}^2(p^c_{j,t}) \\
\text{s.t. } c_1 \leq p^c_{j,t} \leq 0.5.
\end{align*}
\]

IV. SOLVING THE GAME

Note that the optimal strategies of both the attacker and the entities have the same expressions at different time slots. Therefore, the subscript $t$ will be omitted in this section for the ease of presentation. In this work, we focus on the scenario where $q_1(Y_j = 1|s_1) = 0.5$ and $q_2(Y_j = 0|s_1) = 0.5$ for all $j$ without loss of generality.

A. The First-layer Leader-Follower Game

The leader-follower game is often solved by backward induction. First, solve the follower’s problem for every possible strategy taken by the leader. The solution consists of the best response strategy of the follower as a function of the leader’s strategy. Then, the leader decides its optimal strategy according to the followers’ best responses. The obtained solution is often referred to as a Stackelberg-Nash equilibrium (SNE) [16].

By performing backward induction, the best response of entity $i$ can be solved as
\[
p^D(d_1) = \begin{cases} 1 & \text{if } F^i(a_1|Y_i, \hat{Y}_i) > \frac{C_\epsilon}{W}, \\ 0 & \text{if } F^i(a_1|Y_i, \hat{Y}_i) < \frac{C_\epsilon}{W}. \end{cases}
\]

Theorem I: Combining the payoff function of the attacker, the SNE of the attacker and entity $i$ can be obtained as follows:
\[
p^A(a_1) = \begin{cases} \frac{C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}{(2b - C_\epsilon)p(Y_i = 1, \hat{Y}_i = 1|a_1) + C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}, \\ \frac{C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}{(2b - C_\epsilon)p(Y_i = 1, \hat{Y}_i = 1|a_1) + C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}. \end{cases}
\]

Proof: See Appendix A.

Remark I: The SNE obtained above is a weak equilibrium since when $p^A(a_1) = C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)$ for any $p^D(d_1) \in [0,1]$, entity $i$ would receive the same payoff. To push entity $i$ to choose its desired strategy (i.e., $p^D(d_1) = 0$), the attacker will set
\[
p^A(a_1) = \begin{cases} \frac{C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}{(2b - C_\epsilon)p(Y_i = 1, \hat{Y}_i = 1|a_1) + C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}, \\ \frac{C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}{(2b - C_\epsilon)p(Y_i = 1, \hat{Y}_i = 1|a_1) + C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)} - \epsilon, \end{cases}
\] where $\epsilon$ is a small positive number. In this case, the corresponding payoff is only slightly less than the desired SNE obtained above when $\epsilon$ is sufficiently small, which is acceptable for the attacker. For the ease of discussion, $\epsilon$ is set to be 0 in the following analysis, but the results obtained still hold when $\epsilon > 0$, as long as it is sufficiently small.

Remark 2: At the SNE obtained above, the optimal strategy of entity $i$ is to respond with probability $p^c_{j,t}(d_1) = 0$. This is because the attacker is modeled as the leader in the game and thus can take the advantage and choose a strategy to force the entities not to respond. Nonetheless, since both $p(Y_1 = 1, \hat{Y}_i = 1|a_2)$ and $p(Y_1 = 1, \hat{Y}_i = 1|a_1)$ are functions of $q_1(\cdot|s_1), \forall j$ which measure the observation capabilities of the entities, the existence of these collaborative security entities renders the attacker to choose a lower attacking probability. The corresponding payoffs of the attacker and entity $i$ at the above SNE are given as follows:
\[
U^A = \begin{cases} \frac{C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}{(2b - C_\epsilon)p(Y_i = 1, \hat{Y}_i = 1|a_1) + C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2)}, \\ C_\epsilon p(Y_i = 1, \hat{Y}_i = 1|a_2). \end{cases}
\]

Note that in this case, $p(Y_1 = 1, \hat{Y}_i = 1|a_2)$ and $p(Y_1 = 1, \hat{Y}_i = 1|a_2)$ are functions of misreport probabilities $p^c_j, \forall j \in \{1, 2, \cdots, N\} \cap \{j \neq i\}$. 
Proposition 1: The collaboration scheme (i.e., entity $j$ shares $\hat{Y}_j$ with entity $i$) will always give a better payoff for entity $i$.

Proof: Let $f(x) = - \frac{C_i W}{2(b - C_i) x + C}$, then

$$U^D_* = f\left(\frac{p(Y_i = 1, \hat{Y}_{-i} = 1|a_1)}{p(Y_i = 1, \hat{Y}_{-i} = 1|a_2)}\right).$$

Let $\hat{Y}_{-i,j}$ denote the set of obfuscated observations shared by the collaborative entities other than entity $i$ and $j$. Then when entity $j$ does not share its observation with entity $i$, the payoff for entity $i$ is given by

$$U^D_{j,*} = f\left(\frac{p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_1)}{p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_2)}\right).$$

Note that $f(x)$ is an increasing function, which means the sufficient and necessary conditions for $U^D_* > U^D_{j,*}$ is given by

$$p(Y_i = 1, \hat{Y}_{-i} = 1|a_1) > p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_1)$$

$$p(Y_i = 1, \hat{Y}_{-i} = 1|a_2) > p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_2).$$

According to the assumption that each security entity would observe the network state independently, we have

$$p(Y_i = 1, \hat{Y}_{-i} = 1|a_1) = p(Y_i = 1, \hat{Y}_{-i} = 1|a_2)$$

$$p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_1) = p(Y_i = 1, \hat{Y}_{-i,j} = 1|a_2).$$

Therefore, the sufficient and necessary condition for entity $i$ payoff improvement is given by

$$p(\hat{Y}_j = 1|a_1) > p(\hat{Y}_j = 1|a_2).$$

By the assumption that $q_i(Y_j = 1|s_i = 1) > 0.5$, $q_j(Y_j = 0|s_i = 0) > 0.5$ and that $p_j < 0.5$, the above inequality always holds.

B. The Second-layer Game

Given the payoff functions of the entities for all possible collaboration strategies, the payoff at SNE is used as the estimate, and hence $R^{coll}_t(p^c) = U^D_* (p^c)$, and then the utility function of entity $i$ is given by

$$U^{D,2}_j (p^c) = U^D_* (p^c) - U^D_{j,*} (p^c; p_j^c) = 0.5 - \lambda_j P_L (p_j^c).$$

In addition, the action set of entity $j$ is given by $A_j = \{p_j^c | c_j \leq p_j^c \leq 0.5\}$. As a common approach in literature (e.g., [17, 18]), pure strategy Nash Equilibrium is considered for the second-layer game.

Definition 1: [19] A Nash equilibrium $\{p^c_\ast\} = [p_{1\ast}, \cdots, p_{N\ast}]$ for the game is a set of strategies that satisfy

$$U^{D,2}_j (p_j^\ast, p_{-j\ast}) \geq U^{D,2}_j (p_j^c, p_{-j\ast}^c), \forall p_j^c \in A_j, j \in N,$$

in which $p_{-j\ast}^c = \{p^c_k : k \neq j, k \in N\}$ is comprised of the misreport probabilities of other entities.

Theorem 2: ([20]) For each $j \in N$, let $A_j$ be a closed, bounded and convex subset of a finite-dimensional Euclidean space and the payoff function $U^{D,2}_j : A_1 \times A_2 \times \cdots \times A_N \rightarrow \mathbb{R}$ be jointly continuous in all its arguments and strictly concave in $p_j^c \in A_j$ for every $j \in N$. Then the associated $N$-person non-zero-sum game admits a Nash equilibrium in pure strategies.

Given the utility functions and the action sets of all the entities, relying on Theorem 2, we can prove that the second-layer game admits a pure strategy Nash equilibrium (NE) in certain conditions.

Proposition 2: The second-layer game admits a Nash equilibrium in pure strategy when the following condition holds:\footnote{Note that this condition always hold when the network is large enough, i.e., $N \rightarrow \infty$.}

$$A(j) < B(i,j), \forall i, j \in \{1, 2, \cdots, N\} \cap \{j \neq i\},$$

where

$$A(j) = \frac{p(Y_j = 0|a_2) - p(Y_j = 1|a_2)}{p(Y_j = 1|a_1) - p(Y_j = 0|a_1)}$$

$$B(i,j) = \frac{(2b - C_i)p(Y_i = 1|a_1)}{C_i p(Y_i = 1|a_2)} \prod_{k \neq i, j} p(\hat{Y}_k = 1|a_1).$$

Proof: When (17) holds, it can be easily shown that $U^{D,2}_j (p^c)$ is a continuous and strictly concave function of $p_j^c$ (see Appendix B), for $j = 1, 2, \cdots, N$. In addition, the action set is closed, bounded and convex. By theorem 2, the second-layer game admits a Nash equilibrium in pure strategy.

Note that the concavity of the utility function makes problem (11) a convex optimization problem, which is easy to solve numerically. Suppose that all the entities solve the corresponding convex optimization problems asynchronously and broadcast their collaboration strategies using their own timescale. Let $T^d_j$ denote the set of times that entity $j$ updates its misreport probability, and assume that these sets are infinite for all the entities (i.e., all the entities will update infinitely often), an asynchronous dynamic update algorithm is proposed to compute the NE of the second-layer game as in Algorithm 1.

V. BYZANTINE ENTITIES

In the previous discussion, it is assumed that all the collaborative security entities will send their obfuscated observations and misreport probabilities honestly. Nonetheless, in practice, this assumption may not always hold. For example, there may exist some selfish entities which do not want to share their obfuscated observations but they still want to gain benefit from the collaboration scheme. As a result, they generate their obfuscated observations randomly but send out wrong misreport probabilities. Even worse, in the case that some entities are compromised (e.g., taken down by previous attacks), they may send out wrong observations and misreport probabilities deliberately, which will eventually mislead the other entities.
Therefore, for entity $j$, the obfuscation in order for privacy preserving (i.e., state from its own view after obtaining the observation and do the obfuscation in order for privacy preserving (i.e., obfuscation in order for privacy preserving (i.e., state; \( Y_j \)) states. As a result, the objective of the entities is to collaboratively estimate the state of the network, and then choose the optimal responding strategy based on the network state.

Note that given entity $j$’s obfuscated observation \( \hat{Y}_j \) and its misreport probability \( p_j^* \), the distribution of the network state from entity $j$’s view can be obtained as follows:

\[
F_j^i(s|\hat{Y}_j) = \frac{p(\hat{Y}_j|s)p(s)}{p(\hat{Y}_j)},
\]

in which \( p(s) \) is the distribution of the network state determined by the attacking strategy of the attacker; \( p(\hat{Y}_j|s) \) is the distribution of the obfuscated observation \( \hat{Y}_j \) given the network state; \( p(\hat{Y}_j) \) is the distribution of the obfuscated observation \( \hat{Y}_j \). They are given by

\[
p(\hat{Y}_j = 1|s = n) = q_j(\hat{Y}_j = 1|s = n)(1 - p_j^*) + q_n(\hat{Y}_j = 0|s = n)p_j^*, \quad n \in \{0, 1\},
\]

\[
p(\hat{Y}_j = 0|s = n) = q_j(\hat{Y}_j = 0|s = n)(1 - p_j^*) + q_n(\hat{Y}_j = 1|s = n)p_j^*, \quad n \in \{0, 1\},
\]

\[
p(\hat{Y}_j = 0|s = 1) = p(\hat{Y}_j = 0)p(s = 1) + p(\hat{Y}_j = 0)p(s = 0) \quad \text{and} \quad p(\hat{Y}_j = 1|s = 1) = p(\hat{Y}_j = 1)p(s = 1) + p(\hat{Y}_j = 1)p(s = 0).
\]

\[
(23)
\]

\[
(22)
\]

\[
(21)
\]

\[
(20)
\]

With these considerations, it is assumed in this section that at most \( f \) entities may be Byzantine faulty and may behave arbitrarily [21]. In addition to the assumption that all the entities will share their obfuscated observations with entity \( i \), it is further assumed in this section that the entities share their obfuscated observations and misreport probabilities with all the other entities. As a result, the objective of the entities is to collaboratively estimate the state of the network, and then choose the optimal responding strategy based on the network state.

Note that given entity \( j \)’s obfuscated observation \( \hat{Y}_j \) and its misreport probability \( p_j^* \), the distribution of the network state from entity \( j \)’s view can be obtained as follows:

\[
F_j^i(s|\hat{Y}_j) = \frac{p(\hat{Y}_j|s)p(s)}{p(\hat{Y}_j)},
\]

in which \( p(s) \) is the distribution of the network state determined by the attacking strategy of the attacker; \( p(\hat{Y}_j|s) \) is the distribution of the obfuscated observation \( \hat{Y}_j \) given the network state; \( p(\hat{Y}_j) \) is the distribution of the obfuscated observation \( \hat{Y}_j \). They are given by

\[
p(\hat{Y}_j = 1|s = n) = q_j(\hat{Y}_j = 1|s = n)(1 - p_j^*) + q_n(\hat{Y}_j = 0|s = n)p_j^*, \quad n \in \{0, 1\},
\]

\[
p(\hat{Y}_j = 0|s = n) = q_j(\hat{Y}_j = 0|s = n)(1 - p_j^*) + q_n(\hat{Y}_j = 1|s = n)p_j^*, \quad n \in \{0, 1\},
\]

\[
p(\hat{Y}_j = 0|s = 1) = p(\hat{Y}_j = 0)p(s = 1) + p(\hat{Y}_j = 0)p(s = 0) \quad \text{and} \quad p(\hat{Y}_j = 1|s = 1) = p(\hat{Y}_j = 1)p(s = 1) + p(\hat{Y}_j = 1)p(s = 0).
\]

\[
(23)
\]

\[
(22)
\]

\[
(21)
\]

\[
(20)
\]

According to [22], the correctness of Algorithm 2 can be proved when \( N \geq (d + 2)f + 1 \). Interested readers may refer to [22, 23] for more details. Based on the Byzantine Vector Consensus obtained in Algorithm 2, entity \( i \) further estimates the probability of the attacker launching an attack and chooses the optimal responding strategy. The corresponding algorithm is proposed and summarized as in Algorithm 3.

\[
\text{Theorem 4}: \text{Let} \ N_g \text{denote the set of non-faulty entities, then when} \ N \geq (d + 2)f + 1 \text{and} \ F_j(s|\hat{Y}_j) = F_j(s|\hat{Y}_j), \forall i, j \in \ N_g, s \in \{0, 1\}, \text{the only possible decision vector is identical to the input vectors of the non-faulty entities.}
\]
Algorithm 3 Approximate BVC based Algorithm

1. Initialization step: All entities observe the network state, obfuscate the observations after they determine the misreport probability, and estimate the distribution of the network state, set \( v_j(0) = [F^j(s = 0|Y_j), F^j(s = 1|Y_j)], \forall j \).
2. Run Algorithm 2 and obtain the decision vector at Byzantine Vector Consensus \( \hat{v}_j = [\hat{F}^j(s = 0|Y_j), \hat{F}^j(s = 1|Y_j)], \forall j \).
3. Entity \( i \) computes the new \( p(\hat{Y}_j|s), \forall j \) according to (20).
4. Entity \( i \) estimates the probability of the attacker launching an attack (i.e., \( F^i(a_i|Y_{i,t}, \hat{Y}_{i,t}) \)) according to (2).
5. Entity \( i \) determines the optimal responding strategy accordingly.

Proof: According to the Validity condition, the decision vector at each non-faulty entity must be in the convex hull of the input vectors at the non-faulty entities. Since all the input vectors are the same, the convex hull of these input vectors are identical to themselves.

Corollary 1: When \( N \geq (d + 2)f + 1 \) and \( F^i(s|Y_i) = 1 = F^j(s|Y_j), \forall i, j \in \mathcal{N}_g, s \in \{0, 1\} \), the collaborative scheme will suffer no loss from the existence of \( f \) Byzantine entities.

Proof: Note that the optimal attacking strategy of the attacker is given by

\[
p^*_a(a_1) = \frac{C_r}{(2b - C_r) p(Y_i = 1, Y_{-i} = 1|a_1) + C_r} \prod_{j \neq i} p(Y_j = 1|a_1) + C_r
= \left[\frac{2b - C_r}{(2b - C_r) p(Y_i = 1|a_1) + C_r} \prod_{j \neq i} p(Y_j = 1|a_1) + C_r\right]^{(25)}
\]

According to Theorem 4, when \( N \geq (d + 2)f + 1 \) and \( F^i(s|Y_i) = 1 = F^j(s|Y_j), \forall i, j \in \mathcal{N}_g, s \in \{0, 1\} \), the decision vector is the same as the input vectors for all the non-faulty entities. In this case, it can be further verified that \( p(Y_i = 1|a_1) \) remains the same for all \( j \in \mathcal{N}_g \). As a result, in the case that there are \( N - f \) non-faulty entities and \( f \) faulty entities, the optimal attacking strategy of the attacker and the expected payoffs of the entities are the same as those of the case in which there are only \( N - f \) non-faulty entities. Moreover, since the Byzantine entities can behave arbitrarily, they may choose to collaborate with other entities and act like non-faulty entities. In this case, the expected payoffs of the entities may be even higher than that of the case in which there are only \( N - f \) non-faulty entities. In summary, the collaborative scheme will suffer no loss from the existence of \( f \) Byzantine entities.

VI. NUMERICAL STUDY

In this section, numerical study is performed to validate the analytical results.

A. Utility-privacy Tradeoff

In this subsection, we consider a network consisting of \( N \) collaborative security entities, and it is assumed that all of them have high security requirements. In such a scenario, the security entities would have more powerful response capability and the cost of response is considered to be small (i.e., \( b \) is large and \( C_r \) is small). Considering these, we set \( C_a = C_r = 0.1, W = 1000, b = 0.9, q_j(Y_{j,t}|s_t) = 0.7 \) if \( Y_{j,t} = s_t \) and \( q_j(Y_{j,t}|s_t) = 0.3 \), otherwise, for \( j = 1, 2, \ldots, N \).

Fig. 2 shows the tradeoff between the average payoff improvement (i.e., the difference of the utility of the collaborative scheme and that of the non-collaborative case in which no entity shares its observation with entity \( i \)) and the preserved privacy (i.e., \( 1 - P_L(p^*_a) \)) of all the entities. It can be seen that in all the examined scenarios, the collaborative scheme always enhances the performance, which justifies Proposition 1. In addition, the payoff improvement achieves its highest value when the preserved privacy is 0 (i.e., all the entities share their observations with entity \( i \) honestly) and the payoff improvement vanishes to 0 when the preserved privacy attains 1 (i.e., all the entities randomly send out their observations with probability 0.5). Furthermore, when the number of collaborative entities increases, the entities can reserve more privacy while achieving the same the payoff improvement. Intuitively, when there are more collaborative entities, entity \( i \) can gather more information about whether the attacker has launched an attack or not, given all the shared observations. As a result, once the attacker launches an attack, the probability of being detected and triggering the entities to respond is higher, which in turn decreases the attacker’s attacking probability.

B. Collaboration Strategies

In this subsection, the optimal collaboration strategies of entities in the three collaborative entities case are examined (i.e., entity \( j \) and entity \( k \) are sharing their observations with entity \( i \), similar results are observed for the cases of \( N > 3 \)).

Assuming \( \lambda_k = 10 \), Fig. 3 shows how \( \lambda_j \) will influence the collaboration strategies of both entities. In our model, \( \lambda_j \) determines how important privacy is for entity \( j \), and is thus closely related to its privacy requirement. It can be seen that
with different \( \lambda_j \), not only the misreport probability of entity \( j \) changes but also that of entity \( k \). More specifically, when \( \lambda_j \) becomes larger, entity \( j \) would collaborate with higher misreport probability while entity \( k \) would collaborate with lower misreport probability. This may be explained as follows: a larger \( \lambda_j \) implies that entity \( j \) emphasizes more on privacy, and hence it would prefer to increase its misreport probability. In the meantime, with larger \( p_j^c \), the second-layer game transits to another NE point, which corresponds to a smaller \( p_k^c \) due to concavity of the utility function.

In addition, it is worth mentioning that for different privacy requirements (i.e., different \( \lambda_j \) and \( \lambda_k \)), our model is able to guide the entities in finding optimal collaboration strategies that can achieve a suitable balance between utility and privacy.

C. Convergence of the Dynamic Update Algorithm

In this subsection, the convergence of the proposed dynamic update algorithm is examined. Assuming that \( \lambda_j = 7, \lambda_k = 10 \), Fig. 4 shows the misreport probabilities of both entity \( j \) and entity \( k \). It can be seen that the misreport probabilities converge to the NE (similar results can be observed for different \( \lambda_j \) and \( \lambda_k \)), which verifies the effectiveness of Algorithm 1.

D. Byzantine Entities

In this subsection, the influence of Byzantine entities is examined. A network that consists of 5 collaborative security entities is considered. It is assumed that one of the entities is faulty and will always act in favor of the attacker (i.e., it always broadcasts obfuscated observation \( \hat{Y} = 0 \) and claims the misreport probability is \( p^c = 0 \)). One the other hand, the other entities are assumed to have the same privacy requirements (i.e., the same \( \lambda \)) and thus will misreport with the same probability. Fig. 5 shows the payoff improvement in terms of different misreport probabilities. It can be seen that the collaborative scheme with Approximate BVC algorithm performs better than the one without using Approximate BVC algorithm, and it agrees with the case that there are only 4 non-faulty entities. This is because when the non-faulty entities have the same input vectors, the output of the consensus is identical to the input vectors and therefore the faulty entity has no influence on the performance of the collaborative scheme.

VII. Related Works

With the rapid development of sophisticated large-scale attacks, the performance of an individual security system is rarely satisfactory. As a result, various collaborative security models have been proposed in the past two decades. However, only a few of them take the privacy concerns into consideration. In [5] and [6], privacy of sensitive data from the distributed alerts can be partially achieved by the utilization of Bloom filters due to their probabilistic data structure. In [10] and [7] cryptographic methods are used to preserve the sensitive information in intrusion alert data sharing. [9] and [8] proposed the use of entropy guided alert sanitization, where sensitive attributes of the alerts are generalized to high-level concepts to introduce uncertainty into the dataset. Despite all the privacy-aware collaborative security schemes, none of them is amenable to quantitatively study the tradeoff between the utility and the payoff. Different from the works mentioned above, our work studies the utility-privacy tradeoff from a game-theoretic viewpoint and derived the NE and SNE which provide the optimal collaboration as well as response...
strategies of the collaborative security entities for different
privacy requirements.

VIII. CONCLUSIONS AND FUTURE WORKS

In this work, the utility-privacy tradeoff problem in collabora-
tive security is formulated as a repeated two-layer single-leader
multi-follower game which ends once the entities respond to
the attacker successfully. By solving the first layer leader-
follower game, the utility-privacy tradeoff curve for given
collaboration strategies depending on the privacy policies of
different entities is obtained. By solving the second layer
game, the collaborative strategies for the entities at NE can
be computed. In addition, the existence of NE of the second-
layer game is proved and an asynchronous dynamic update
algorithm is developed to compute the NE. The existence of
Byzantine entities is also considered and studied. Further
extending this work to dynamic settings or multiple possible
attacks settings constitute interesting future directions.

APPENDIX

A. Proof of Theorem 1

According to (26),

\[ U_t^A(p_t^A, p_t^D(p_t^A)) = \sum_{Y_{i,t}, \tilde{Y}_{-i,t} \in \{0,1\}^N} [p(Y_{i,t}, \tilde{Y}_{-i,t})p_t^A(a_1)p_t^D(d_1|Y_{i,t}, \tilde{Y}_{-i,t})(W - C_a W) + p(Y_{i,t}, \tilde{Y}_{-i,t})p_t^A(a_1)p_t^D(d_1|Y_{i,t}, \tilde{Y}_{-i,t}))[1 - 2b(C_a W)], \]

where \( p_t^D(d_1) \) is given by

\[ p_t^D(d_1) = \begin{cases} 1 & \text{if } F^a(a_1|Y_{i,t}, \tilde{Y}_{-i,t}) > \frac{C_a}{2b}, \\ \in [0,1] & \text{if } F^a(a_1|Y_{i,t}, \tilde{Y}_{-i,t}) = \frac{C_a}{2b}, \\ 0 & \text{if } F^a(a_1|Y_{i,t}, \tilde{Y}_{-i,t}) < \frac{C_a}{2b}. \end{cases} \]

The attacker finds its optimal strategy by solving the following
optimization problem:

\[ p_t^A(p_t^D) = \arg\max_{p_t^A} \sum_{t=1}^{T_e} U_t^A(p_t^A, p_t^D(p_t^A)). \]  

1) Case 1: If the attacker chooses its strategies \( p_t^A(a_1) = p_t^{A(*)}(a_1) \) such that \( p_t^D(d_1) = 0, \forall t \), which means the best
response of entity \( i \) is to choose “do nothing”, the expected
total payoff of the attacker (denoted by \( U(0) \)) is given by

\[ U(0) = \sum_{t=1}^{T_e} U_t^A(p_t^A, p_t^D(p_t^A)) = T_e p_t^{A(*)}(a_1)[1 - C_a W], \]

where \( T_e = \infty \).

2) Case 2: If the strategies chosen by the attacker are as
follows

\[ p_t^A(a_1) = \begin{cases} p_t^{A(*)}(a_1) & \text{if } t \notin T_K, \\ p_t^A(a_1) & \text{Otherwise}. \end{cases} \]

such that

\[ p_t^D(d_1) = \begin{cases} 0 & \text{if } t \notin T_K, \\ \in [0,1] & \text{Otherwise}. \end{cases} \]

where \( T_K = \{t_1, \ldots, t_K\} \) is the set of time when the attacker
chooses its strategies \( p_t^A(a_1) \) such that \( p_t^D(d_1) > 0, \forall t \in T_K \)
and \( t_m < t_n, \forall m < n \).

\( K = 1 \): First of all, considering the case that \( K = 1 \), the
expected total payoff of the attacker (denoted by \( U(1) \)) is given by

\[ U(1) = \sum_{t=1}^{T_e} U_t^A(p_t^A, p_t^D(p_t^A)) = bp_t^{A1}(a_1)p_t^D(d_1)G(t_1) + \sum_{t=1}^{t_1-1} [p_t^{A1}(a_1)[1 - C_a W] + \sum_{t=t_1+1}^{T_e} p_t^{A1}(a_1)[1 - C_a W], \]

where \( bp_t^{A1}(a_1)p_t^D(d_1) \) is the probability that the attacker
launches an attack and entity \( i \) responds to the attack success-
fully and \( G(t_1) \) is the payoff of the attacker obtained at time
\( t_1 \) which is bounded (i.e., \( G(t_1) < \infty \). Note that the closed-
form of \( G(t_1) \) can also be obtained, although it is unnecessary
here as long as it is bounded. Then

\[ U(0) - U(1) = bp_t^{A1}(a_1)p_t^D(d_1) \sum_{t=1}^{T_e} p_t^{A1}(a_1)[1 - C_a W] + \sum_{t=t+1}^{T_e} p_t^{A1}(a_1)[1 - C_a W] - bp_t^{A1}(a_1)p_t^D(d_1)G(t_1). \]

Since \( T_e = \infty \) and \( G(t_1) < \infty \), we have \( U(0) > U(1) \).

\( K > 1 \): Assuming that we now have \( K = k \) where
\( k \geq 1 \), if the attacker further chooses to launch an attack
with probability \( p_{t+k+1}^A(a_1) \) such that \( p_{t+k+1}^D(d_1) > 0 \) at time
\( t_{k+1} > t_k \). The expected payoffs of the attacker when \( t < t_{k+1} \)
will remain the same, but the total expected payoffs of the
attacker when \( t \geq t_{k+1} \) will decrease according to the
discussion above. As a result, \( U(k) > U(k+1), \forall k \geq 1 \).
Therefore

\[ U(0) > U(k), \forall k > 0, \]

which means the optimal strategy of the attacker is to choose
the maximum \( p_t^A(a_1) = p_t^{A(*)}(a_1) \) such that \( p_t^D(d_1) = 0, \forall t \).

As a result, the corresponding SNE is given by

\[ \begin{cases} p_t^A(a_1) = \frac{C_p p(Y_{i,t}=1, \tilde{Y}_{-i,t}=1|a_2)}{(2b - C_p) p(Y_{i,t}=1, \tilde{Y}_{-i,t}=1|a_2) + C_p}, \\ p_t^D(d_1) = 0. \end{cases} \]

B. Concavity of second-layer game utility function

According to the discussion in Section IV,

\[ U_t^D(p_t^f) = \frac{-C_w}{(2b - C_r) p(Y_{i,t}=1|a_1) p(Y_{j,t}=1|a_2) + C_r}, \]

where

\[ \begin{align*} &((2b - C_r) p(Y_{i,t}=1|a_1) p(Y_{j,t}=1|a_2) + C_r, \\
&= \left[ \begin{array}{c} p(Y_{i,t}=1|a_1) p(Y_{j,t}=1|a_2) + C_r, \end{array} \right]. \end{align*} \]
where
\[
\frac{p(Y_j = 1|a_1)}{p(Y_j = 1|a_2)} = \frac{p(Y_j = 1|a_1)(1 - p_j^c) + p(Y_j = 0|a_1)p_j^c}{p(Y_j = 1|a_2)(1 - p_j^c) + p(Y_j = 0|a_2)p_j^c}.
\]

Let
\[
a = C_r p(Y_j = 1|a_2),
\]
\[
b = C_r p(Y_j = 0|a_2) - p(Y_j = 1|a_2),
\]
\[
m = [(2b - C_r)\frac{p(Y_j = 1|a_1)}{p(Y_j = 1|a_2)} \prod_{k \neq i,j} p(\hat{Y}_k = 1|a_1)] - p(Y_j = 1|a_1) + C_r p(Y_j = 1|a_2),
\]
\[
n = [(2b - C_r)\frac{p(Y_j = 1|a_1)}{p(Y_j = 1|a_2)} \prod_{k \neq i,j} p(\hat{Y}_k = 1|a_1)] - p(Y_j = 0|a_1) - C_r p(Y_j = 0|a_2).
\]

Then, (32) can be expressed as
\[
U_D^*(p_j^c) = \frac{-a + bp_j^c}{m + np_j^c} W.
\]

By the assumption that \(2b - C_r > 0, q_j(Y_j = 1|s_t = 1) > 0.5, q_j(Y_j = 0|s_t = 0) > 0.5\) and \(p_j^c < 0.5\) for all \(j, a > 0, b > 0,\) and \(m > 0.\) Therefore, the concavity of \(U_D^*(p_j^c)\) is determined by \(n.\)

1) Case 1: \(n = 0:\) In this case, \(U_D^*(p_j^c) = \frac{-a + bp_j^c}{m} W.\) Apparently, it is concave but not strictly concave in terms of \(p_j^c.\)

2) Case 2: \(n \neq 0:\) In this case, (37) can be expressed as
\[
U_D^*(p_j^c) = -\frac{b}{n} + \frac{a - bm}{m + p_j^c} W = -\frac{b}{n} + \frac{1}{n^2} \frac{an - bm}{m + p_j^c} W,
\]
where \(an - bm < 0.\) Therefore, the sufficient and necessary condition for \(U_D^*(p_j^c)\) being strictly concave when \(p_j^c \in [c_j, 0.5]\) is given by
\[
m \frac{n}{n} < -\frac{1}{2},
\]
which is equivalent to
\[
A(j) < B(i, j),
\]
where
\[
A(j) = \frac{p(Y_j = 0|a_2) - p(Y_j = 1|a_2)}{p(Y_j = 1|a_1) - p(Y_j = 0|a_1)}
\]
\[
B(i, j) = \frac{(2b - C_r)p(Y_j = 1|a_1)}{C_r p(Y_j = 1|a_2) \prod_{k \neq i,j} p(\hat{Y}_k = 1|a_1)}.
\]

Note that given fixed \(p_{e-j}^c, U_D^*(p_{e-j}^c, p_j^c = 0.5)\) is constant. Furthermore, \(-\lambda_j P_{i, j}(p_j^c)\) is also strictly concave by its definition in (10). Therefore, \(U_{D,2}^*(p_j^c)\) is a continuous and strictly concave function of \(p_j^c\) when (40) holds.

References
