Towards Privacy-Aware Collaborative Security: A Game–Theoretic Approach

Richeng Jin*, Xiaofan He†, Huaiyu Dai*, Rudra Dutta‡, Peng Ning§,
*Department of ECE, North Carolina State University, emails: {rjin2,hdai}@ncsu.edu
†Department of EE, Lamar University, email: xhe1@lamar.edu
‡Department of CSC, North Carolina State University, email: dutta@ncsu.edu
§Samsung Research America, email: peng.ning@samsung.com

***Abstract*—With the rapid development of sophisticated attack techniques, individual security systems that 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 in which a set of security entities are coordinated to perform specific security actions is proposed in literature. In collaborative security schemes, multiple entities collaborate with each other by sharing threat evidence or analytics to make more effective decisions. Nevertheless, the anticipated 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) for the collaborative entities is proven, 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 presented to validate the analysis.**

I. Introduction

Individual security systems mainly rely on their own (often limited) observations and knowledge to make security decisions and take actions to prevent and respond to attacks. 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 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, insider attacker identification and botnet detection (see, e.g., [2, 3] and the references therein). The objective of collaborative security is to enhance security performance through strategically sharing security-related information with each other. Considering that the entities in collaborative security are often independent and hence may take self-interested actions, game theory has been widely employed to devise the collaboration strategies in various collaborative security applications [4–9]. In these works, the entities devote some efforts to exploring the vulnerabilities in a common "platform" and share the security-related information for a better defense. However, the game-theoretic analysis and the corresponding strategy design in these works only concern the interactions among the collaborative entities. In practice, the behaviors of the attackers usually greatly affect the strategies of the entities. For example, when attacks are launched on certain entities, they will have a stronger desire to increase security investment and gather information from the collaborators. In addition, the privacy issue that an entity's private information may be leaked in the information sharing process has been largely ignored. In practice, if not properly addressed, such privacy concern may deter entities from collaborating.

Some techniques have been proposed in literature to protect the privacy [10–18], at the cost of utility loss (i.e., a degradation in collaboration effectiveness). 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 security entities and the intelligence of the attacker, a game-theoretic approach is taken in this work. More specifically, the interaction between 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, based on which the corresponding utility-privacy tradeoff curve is obtained. The second-layer focuses on the interactions among security entities themselves, 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

This work was supported in part by the NCSU Science of Security Lablet, and in part by National Science Foundation under Grants ECCS-1307949 and EARS-1444009.

978-1-5386-1027-5/17/$31.00 ©2017 IEEE

proposed two-layer game model is presented in Section III. The proposed game is solved in Section IV. The impact of Byzantine entities and corresponding solutions are 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 $\mathcal{N} = \{1, 2, ..., N\}$. Let s_t denote the state of the network at time t.

A. Attacker Model

An external 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 its goal is to maintain the attack on the network as long as possible. In each time slot, it is assumed that the attacker will receive an instant reward if it launches an attack successfully without being identified. The follow-up actions are beyond the scope of this paper.

Furthermore, for the ease of presentation, the following discussion will be focused on 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.

The action space of the attacker against the network is $\mathcal{A} = \{a_1, a_2\}$, 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 $\boldsymbol{p_t^A} = [p_t^A(a_1), p_t^A(a_2)]$, 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. The game between the attacker and defender is repeated until the attack is identified and addressed successfully by the security entities. If the attack is identified and addressed successfully, the attacker will stop using the same type of attack (in the time frame of interest). This assumption makes sense because an attacker usually launches an attack by exploiting the vulnerabilities of the system and once the attack is detected, 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 starting a new game in our model, which is hence not considered here for simplicity.

B. Defender Model

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

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 entities in the network can collaborate and share their observations so as to further enhance the network security. However, considering that the observations are private, such observation sharing will lead to potential privacy leakage for the entities. In order to preserve privacy, 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 with probability $p^c_{j,t}$, which is assumed publicly known. The preserved privacy is measured by the entropy induced by $p^c_{j,t}$ [19], given as follows:

$$H(p^c_{j,t}) = -p^c_{j,t} \log_2(p^c_{j,t}) - (1 - p^c_{j,t}) \log_2(1 - p^c_{j,t}). \quad (1)$$

In addition, it is further assumed that all the collaborative entities will elect a trustworthy master entity that acts as the defender of the network, which may be rotated from time to time. The defender will collect the shared observations, and suggest a recommended action for all entities to follow. To this end, the objective of the defender is to respond to the attacks properly on behalf of all the entities when the network is under attack. Its action space is $\mathcal{D} = \{d_1, d_2\}$, where d_1 corresponds to "respond" and d_2 corresponds to "do nothing". The mixed strategy chosen by the defender is denoted by $\boldsymbol{p}_t^D = [p_t^D(d_1), p_t^D(d_2)]$, in which $p_t^D(d_1)$ and $p_t^D(d_2)$ are the probabilities that it takes action d_1 and d_2 at time t, respectively. Note that since all entities will follow the recommended action of the defender, the action and payoff of the defender are used to represent those of the whole network in the following discussion.

C. Payoff Settings

Let W denote the loss of security when an attack is successfully launched. In this case, it is assumed without loss of generality that the attacker gets a payoff W and the defender gets a payoff $-W$.[2] In contrast, if the attack is detected and successfully addressed, the payoffs for the attacker and the defender are assumed to be $-W$ and W, respectively. Table I illustrates the payoff matrix of the attacker/defender interaction, in which the first entry and second entry in each cell denote the payoffs of the attacker and the defender, respectively. In the matrix, $b \in [0, 1]$ denotes the possibility of successful response to the attack, which depends on the responding capability of the entities and is independent of their observation capability (i.e., $\mathcal{Q}^j$). Similar to [20], the cost of attacking and responding are assumed to be proportional to W, denoted by C_aW and C_rW, respectively, in which C_a and C_r denote the corresponding cost coefficients. The first cell

[1] When the attacker launches multiple types of attacks independently, multiple independent games can be formed, each corresponding to a different type of attack. For the case that the attacker combines the efforts of multiple attacks to improve the successful rate, it can be considered as one type of attack.

[2] The model can be readily generalized into the case that the sum of payoffs of the attacker and the entities are not equal to 0.

corresponds to the case when the attacker chooses to attack and the defender chooses to respond at the same time. Since the probability of successful response for the network is b, the defender will get payoff $W - C_rW$ and $-W - C_rW$ with probability b and $1 - b$, respectively. Therefore, the expected payoff of the defender is $-(1-2b)W - C_rW$, while that of the attacker is $(1-2b)W - C_aW$. The payoffs of both the attacker and the defender in other cases can be obtained similarly. Note that when the defender chooses to "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, $C_r < 1$.

D. A Concrete Example

To help understand the models mentioned in the previous subsections, an example is provided as follows.

Example. *In this example, at each time slot t, the attacker can choose to launch a DDoS attack on the whole network. When it launches the attack, the network state becomes abnormal (i.e., $s_t = 1$). As a result, the network resource becomes unavailable to the entities in the network and the corresponding services are disrupted. Therefore, the entities in the network will suffer a loss of W, which is determined by the specific services or amount of resource being compromised. Intuitively, to produce more detrimental effects, the attacker has to spend more effort. Therefore, the cost is set to be proportional to W and denoted by C_aW.*

The entities in the network will deploy their own IDSs to monitor the network traffic as a first line of defense. When an attack is launched, the network state becomes abnormal (i.e., $s_t = 1$), and the IDSs output the detection results $\{Y_{j,t}\}_{j=1}^N$. 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 $q_j(Y_{j,t} = 0|s_t = 1)$ correspond to the detection rate and false negative rate of IDS j, respectively, while $q_j(Y_{j,t} = 1|s_t = 0)$ represents the false positive rate.

Due to possible false alarms and missed detections, the detection results of an individual entity may not be sufficient to decide whether there is a DDoS attack in the network or not. To make more effective decisions, the entities in the network will elect a trustworthy entity to act as the defender and share their detection results with him. However, sharing the detection results may lead to potential privacy leakage for the entities.[3] *As a result, the entities will share an obfuscated version of the detection results. Based on the shared detection results, the defender can evaluate the possibility of DDoS attack and decide whether to invest some effort to identify and track the attacker, at a cost of C_rW. If the defender identifies the attacker successfully, the network will recover from the damage caused by the attack and the defender will receive a reward W. Successful identification of the attacker can also help prevent the attacks from the same attacker in the future.*

TABLE I
PAYOFF MATRIX OF THE GAME

	Respond	Do nothing
Attack	$(1 - 2b)W - C_aW$, $-(1 - 2b)W - C_rW$	$W - C_aW$, $-W$
No attack	0, $-C_rW$	0, 0

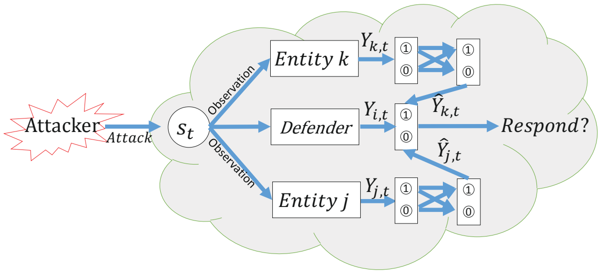

Fig. 1. Diagram of the game model.

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 which are informed of the attacker's attacking strategy. The first-layer game models the interaction between the attacker and the defender, while the second-layer game models the collaborative information sharing among the entities themselves. Fig. 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 the defender is solved, which determines the optimal payoffs of both the attacker and the defender 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, since it is not possible for the entities to foresee the strategies of the attacker and the future observations of themselves, the best response to the attacker's strategy at each time t is actually the best strategy that an entity can take.

1) The Followers' Problem: Without loss of generality, assume that entity i is elected as the defender. At time t, let $\boldsymbol{\hat{Y}_{-i,t}}$ denote the set of obfuscated observations shared by other entities. Given the attacker's strategy $\boldsymbol{p_t^A}$ and its own observation $Y_{i,t}$, the defender first estimates the probability that the attacker actually launches an attack, which is given by

$$F^i(a_1|Y_{i,t}, \boldsymbol{\hat{Y}_{-i,t}}) = \frac{p_t^A(a_1)p(Y_{i,t}, \boldsymbol{\hat{Y}_{-i,t}}|a_1)}{p(Y_{i,t}, \boldsymbol{\hat{Y}_{-i,t}})}, \tag{2}$$

[3] For example, if entity j detects the attack on the network successfully, it indicates that entity j is aware of the DDoS attack in the network. By eavesdropping the shared information, the attacker can infer the security state of the corresponding entity and therefore design better attacking strategies targeting this specific entity. As another example, an intrusion alert usually contains some private information, such as IP address and processing time, the improper use of which may raise severe privacy concerns for the entities.

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

$$p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}|a_1) = q_i(Y_{i,t}|s_t = 1) \prod_{j \neq i} p(\hat{Y}_{j,t}|s_t = 1), \tag{3}$$

$$\begin{aligned} &p(\hat{Y}_{j,t} = 1|s_t = 1) = \\ &q_j(Y_{j,t} = 1|s_t = 1)(1 - p^c_{j,t}) + q_j(Y_{j,t} = 0|s_t = 1)p^c_{j,t}, \end{aligned} \tag{4}$$

$$\begin{aligned} &p(\hat{Y}_{j,t} = 0|s_t = 1) = \\ &q_j(Y_{j,t} = 0|s_t = 1)(1 - p^c_{j,t}) + q_j(Y_{j,t} = 1|s_t = 1)p^c_{j,t}, \end{aligned} \tag{5}$$

$$\begin{aligned} &p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = \\ &p^A_t(a_1)p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}|a_1) + p^A_t(a_2)p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}|a_2), \end{aligned} \tag{6}$$

in which $q_i(Y_{i,t}|s_t = 1)$ is the probability that the defender observes $Y_{i,t}$ when $s_t = 1$ and $p(\hat{Y}_{j,t}|s_t = 1)$ is the probability that entity j shares $\hat{Y}_{j,t}$ with the defender when $s_t = 1$.

Then, the defender finds its optimal strategy by solving the following optimization problem:

$$\boldsymbol{p^D_t}(\boldsymbol{p^A_t}, Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = \underset{\boldsymbol{p^D_t}}{\text{argmax}} \quad \boldsymbol{U^D_t}(\boldsymbol{p^D_t}, \boldsymbol{p^A_t}, Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}). \tag{7}$$

The payoff function $\boldsymbol{U^D_t}(\boldsymbol{p^D_t}, \boldsymbol{p^A_t}, Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ in (7) is given by

$$\begin{aligned} &\boldsymbol{U^D_t}(\boldsymbol{p^D_t}, \boldsymbol{p^A_t}, Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = \\ &- F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})W \\ &+ F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})[-(1 - 2b)W - C_rW] \\ &- F^i(a_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})C_rW, \end{aligned} \tag{8}$$

where $F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is the probability of the case that the attacker launches an attack and the defender chooses to respond given the observations $Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}$, and $-(1 - 2b)W - C_rW$ is the payoff of the defender in this case; Similarly, $F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is the probability of the case that the attacker launches an attack and the defender chooses to do nothing given the observations $Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}$, and $-W$ is the payoff of the defender in this case; Finally, $F^i(a_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is the probability of the case that the attacker does not launch an attack and the defender chooses to respond given the observations $Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}$, and $-C_rW$ is the payoff of the defender 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 accordingly. However, since the attacker does not know the actual observations of the entities, it has to maximize the expected payoff with respect to the distribution $p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$, which is given by (6).[4] As a result, the attacker finds its optimal strategy by solving the following optimization problem:

$$\boldsymbol{p^A_t}(\boldsymbol{p^D_t}) = \underset{\boldsymbol{p^A_t}}{\text{argmax}} \sum_{t=1}^{T_e} \boldsymbol{U^A_t}(\boldsymbol{p^A_t}, \boldsymbol{p^D_t}(\boldsymbol{p^A_t})), \tag{9}$$

where T_e is the time when the defender successfully responds to the attacker and $\boldsymbol{U^A_t}(\boldsymbol{p^A_t}, \boldsymbol{p^D_t}(\boldsymbol{p^A_t}))$ is given by

$$\begin{aligned} &\boldsymbol{U^A_t}(\boldsymbol{p^A_t}, \boldsymbol{p^D_t}(\boldsymbol{p^A_t})) = \sum_{Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}} \in \{0,1\}^N} \\ &\big[p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^A_t(a_1)p^D_t(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})(W - C_aW) \\ &+ p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})p^A_t(a_1)p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})[(1 - 2b - C_a)W]\big], \end{aligned} \tag{10}$$

where $p^A_t(a_1)p^D_t(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is the probability of the case that the attacker launches an attack and the defender chooses to do nothing given the observations $Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}$ and $W - C_aW$ is the payoff of the attacker in this case, while $p^A_t(a_1)p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is the probability of the case that the attacker launches an attack and the defender chooses to respond given the observations $Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}$ and $(1-2b-C_a)W$ is the payoff of the attacker in this case. Note that since the attacker can also obtain the defender's optimal strategy by solving (7), both $p^D_t(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ and $p^D_t(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ are available to her.

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 $p^c_{j,t} \in [c_j, 0.5]$[5] with which the entity j would send out the wrong observation result in order to protect its own privacy, and c_j depends on the privacy policy of each entity j. The utility function of each entity j is given as follows:

$$\boldsymbol{U^{D,2}_{j,t}}(\boldsymbol{p^c_t}) = R^{est}_{i,t}(\boldsymbol{p^c_t}) - R^{est}_{i,t}(\boldsymbol{p^c_{-j,t}}, p^c_{j,t} = 0.5) - \lambda_j P_L(p^c_{j,t}), \tag{11}$$

where $\boldsymbol{p^c_t} = (p^c_{1,t}, p^c_{2,t}, \cdots, p^c_{N,t})$ is a vector which denotes the misreport probabilities of all the entities; $\boldsymbol{p^c_{-j,t}}$ denotes the misreport probabilities of all the entities other than entity j; $R^{est}_{i,t}(\boldsymbol{p^c_t})$ denotes the estimated payoff of the defender given $\boldsymbol{p^c_t}$, which will be discussed in Section IV; $R^{est}_{i,t}(\boldsymbol{p^c_{-j,t}}, p^c_{j,t} = 0.5)$ denotes the estimated reward of the defender when entity j randomly reports its detection result (i.e., $p^c_{j,t} = 0.5$), and therefore $R^{est}_{i,t}(\boldsymbol{p^c_t}) - R^{est}_{i,t}(\boldsymbol{p^c_{-j,t}}, p^c_{j,t} = 0.5)$ measures the defender's estimated payoff improvement due to the shared observations from entity j; λ_j is a constant that measures the importance of privacy loss, given by

$$P_L(p^c_{j,t}) = 1 - H(p^c_{j,t}), \tag{12}$$

[4] In this work, it is assumed that the attacker is empowered with the knowledge of $\mathcal{Q}^j, \forall j \in \mathcal{N}$, to give a conservative analysis.

[5] In this work, it is assumed that the misreport probabilities are common knowledge for all the entities. Therefore, it is equivalent for an entity to misreport with probability $p^c_{j,t}$ or $1 - p^c_{j,t}$.

where $H(p^c_{j,t})$ denotes the entropy induced by $p^c_{j,t}$ given in (1). As a result, each entity j has to solve the following optimization problem:

$$\begin{aligned} \max_{p^c_{j,t}} \quad & \boldsymbol{U^{D,2}_{j,t}}(\boldsymbol{p^c_t}) \\ \text{s.t.} \quad & c_j \leqslant p^c_{j,t} \leqslant 0.5. \end{aligned} \tag{13}$$

IV. Solving the game

Note that the optimal strategies of both the attacker and the defender 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 $q_j(Y_j = 1|s_t = 1) > q_j(Y_j = 1|s_t = 0)$ for all j without loss of generality.[6]

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 follower's best responses. The obtained solution is often referred to as a Stackelberg-Nash equilibrium (SNE) [21].

Theorem 1. *By performing backward induction, the best response of the defender can be obtained as*

$$p^D(d_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) = \begin{cases} 1 & \text{if } F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) > \frac{C_r}{2b}, \\ \in [0,1] & \text{if } F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) = \frac{C_r}{2b}, \\ 0 & \text{if } F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) < \frac{C_r}{2b}. \end{cases} \tag{14}$$

Proof: According to (8), the payoff function of the defender is given by

$$\begin{aligned} &\boldsymbol{U^D}(\boldsymbol{p^D}, \boldsymbol{p^A}, Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) = -F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}})W \\ &\quad + p^D(d_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}})[2bF^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) - C_r]W. \end{aligned} \tag{15}$$

Therefore, when $F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) > \frac{C_r}{2b}$, it is an increasing function of $p^D(d_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}})$; when $F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) < \frac{C_r}{2b}$, it is a decreasing function of $p^D(d_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}})$; when $F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}) = \frac{C_r}{2b}$, it is a constant function. The best response of the defender is given as (14). ■

Theorem 2. *Combing the payoff function of the attacker, the SNE of the attacker and the defender can be obtained as follows:*

$$\begin{cases} p^A_*(a_1) = \frac{C_r p(Y_i=1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)}{(2b-C_r)p(Y_i=1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_1)+C_r p(Y_i=1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a2)}. \\ p^D_*(d_1) = 0. \end{cases}$$

Proof: See Appendix A. ■

Corollary 1. *With the addition of one more collaborative entity, the attacker's optimal attacking strategy (probability) $p^A_*(a_1)$ decreases; with any one of the collaborative entities increasing its misreport probability, the attacker's optimal attacking strategy (probability) $p^A_*(a_1)$ increases.*

Proof: Let $\mathcal{N}_c \subseteq \mathcal{N}$ denote an arbitrary set of collaborative entities including the defender and $\hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}}$ denote the set of obfuscated observations shared by the collaborative entities other than the defender. Let $f(x) = \frac{C_r}{(2b-C_r)x+C_r}$, then

$$p^A_*(a_1) = f(\frac{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}|a_1)}{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}|a_2)}). \tag{16}$$

When one more entity (denoted as entity j) chooses to join the collaboration, the optimal attacking strategy of the attacker is given by

$$\hat{p}^A_*(a_1) = f(\frac{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}, \hat{Y}_j = 1|a_1)}{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}, \hat{Y}_j = 1|a_2)}). \tag{17}$$

Note that $f(x)$ is a decreasing function of x, which means the sufficient and necessary conditions for $\hat{p}^A_*(a_1) < p^A_*(a_1)$ is given by

$$\frac{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}, \hat{Y}_j = 1|a_1)}{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}, \hat{Y}_j = 1|a_2)} > \frac{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}|a_1)}{p(Y_i = 1, \hat{\boldsymbol{Y}}^{\boldsymbol{\mathcal{N}_c}}_{\boldsymbol{-i}} = \boldsymbol{1}|a_2)},$$

which is equivalent to

$$\frac{p(\hat{Y}_j = 1|a_1)}{p(\hat{Y}_j = 1|a_2)} > 1, \tag{18}$$

in which

$$\frac{p(\hat{Y}_j = 1|a_1)}{p(\hat{Y}_j = 1|a_2)} = \frac{q_j(Y_j = 1|a_1)(1 - p^c_j) + q_j(Y_j = 0|a_1)p^c_j}{q_j(Y_j = 1|a_2)(1 - p^c_j) + q_j(Y_j = 0|a_2)p^c_j}.$$

It can be shown that when $q_j(Y_j = 1|a_1) = q_j(Y_j = 1|s_t = 1) > q_j(Y_j = 1|a_2) = q_j(Y_j = 1|s_t = 0)$ and $p^c_j < 0.5$, (18) always holds; furthermore, $\frac{p(\hat{Y}_j=1|a_1)}{p(\hat{Y}_j=1|a_2)}$ is a decreasing function of p^c_j. Therefore, with the addition of one more collaborative entity, the optimal attacking probability monotonically decreases. On the other hand, with larger p^c_j, the optimal attacking probability monotonically increases. When $p^c_j = 0.5$, it can be shown that $\frac{p(\hat{Y}_j=1|a_1)}{p(\hat{Y}_j=1|a_2)} = 1$ and no collaboration gain can be obtained from entity j. ■

Remark 1. *The SNE obtained above is a weak equilibrium since when $p^A(a_1) = p^A_*(a_1)$, for any $p^D(d_1) \in [0,1]$, the defender will receive the same payoff. To push the defender to choose its desired strategy (i.e., $p^D_*(d_1) = 0$) so that it can achieve higher gain, the attacker will set*

$$p^A(a_1) = p^A_*(a_1) - \epsilon,$$

where ϵ is a small positive number. In this case, the corresponding payoff is only slightly less than the desired SNE obtained above when ϵ is sufficiently small, which is acceptable for the attacker. For the ease of discussion, ϵ 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.

[6] It is reasonable to assume that the entities are equipped with good sensing capabilities. Alternatively, if it is known a prior that some entity is equipped with bad sensors, one may reverse the detection results.

Remark 2. *At the SNE obtained above, the optimal strategy of the defender is to respond with probability $p^D_*(d_1) = 0$. 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, as is shown in Corollary 1, the existence of these collaborative entities renders the attacker to choose a lower attacking probability. The more the collaborative entities, the less likely an attack will be launched. It is also noticed that there is a tradeoff between collaboration utility and privacy preservation, concerning the choice of the obfuscation probabilities.*

The corresponding payoffs of the attacker and the defender at the above SNE are given as follows:

$$\begin{cases} \boldsymbol{U^A_*} = \frac{C_r p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)(1-C_a)W}{(2b-C_r)p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_1)+C_r p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)}. \\ \boldsymbol{U^D_*} = -\frac{C_r p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)W}{(2b-C_r)p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_1)+C_r p(Y_i=1,\hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)}. \end{cases}$$

Note that both utility functions are functions of misreport probabilities $p^c_j, \forall j \in \{1, 2, \cdots, N\} \cap \{j \neq i\}$.

Corollary 2. *Collaboration always decreases the payoff for the attacker and increases the payoff for the defender.*

Proof: It can be seen that

$$\boldsymbol{U^A_*} = (1 - C_a)p^A_*(a_1)W, \tag{19}$$

and

$$\boldsymbol{U^D_*} = -p^A_*(a_1)W. \tag{20}$$

By Corollary 1, with more collaborative entities, $p^A_*(a_1)$ decreases. ■

B. The Second-layer Game

Recall that in Section III-B, the utility function of each entity j is designed as a function of the estimated payoff of the defender. As discussed earlier, it is assumed that the attacker has knowledge about the collaboration strategies of all the entities, and follows the optimal attacking strategy given by the SNE of the first-layer game. Therefore, the payoff of the defender at SNE is used as the estimate, i.e., $R^{est}_i(\boldsymbol{p^c}) = \boldsymbol{U^D_*}(\boldsymbol{p^c})$, and the utility function of entity j is given by

$$\boldsymbol{U^{D,2}_j}(\boldsymbol{p^c}) = \boldsymbol{U^D_*}(\boldsymbol{p^c}) - \boldsymbol{U^D_*}(\boldsymbol{p^c_{-j}}, p^c_j = 0.5) - \lambda_j P_L(p^c_j). \tag{21}$$

In addition, the action set of entity j in the second-layer game is given by $A_j = \{p^c_j | c_j \leqslant p^c_j \leqslant 0.5\}$. As a common approach in literature (e.g., [22, 23]), pure strategy NE is considered here.

Definition 1. *[24] An NE $\{\boldsymbol{p^c_*}\} = [p^c_{1,*}, \cdots, p^c_{N,*}]$ for the game is a set of strategies that satisfy*

$$U^{D,2}_j(p^c_{j,*}, \boldsymbol{p^c_{-j,*}}) \geq U^{D,2}_j(p^c_j, \boldsymbol{p^c_{-j,*}}), \forall p^c_j \in A_j, j \in \mathcal{N}, \tag{22}$$

in which $\boldsymbol{p^c_{-j,*}} = \{p^c_{k,*} : k \neq j, k \in \mathcal{N}\}$ is comprised of the misreport probabilities of all the other entities except entity j.

Theorem 3. *([24]) For each $j \in \mathcal{N}$, let A_j be a closed, bounded and convex subset of a finite-dimensional Euclidean space and the payoff function $U^{D,2}_j : A_1 \times A_2 \cdots \times A_N \to \mathbb{R}$ be jointly continuous in all its augments and strictly concave in $p^c_j \in A_j$ for every $j \in \mathcal{N}$. Then the associated N-person non-zero-sum game admits an NE in pure strategies.*

Given the utility functions and the action sets of all the entities, relying on Theorem 3, we can prove that the second-layer game admits a pure strategy NE under certain conditions.

Proposition 1. *The second-layer game admits an NE in pure strategy when the following condition holds:*[7]

$$\begin{cases} A(j) \leq B(i, j), & \textit{if } \lambda_j > 0, \forall j \in \{1, 2, \cdots, N\} \cap \{j \neq i\}, \\ A(j) < B(i, j), & \textit{if } \lambda_j = 0, \forall j \in \{1, 2, \cdots, N\} \cap \{j \neq i\}, \end{cases} \tag{23}$$

where

$$A(j) = \frac{p(Y_j = 0|a_2) - p(Y_j = 1|a_2)}{p(Y_j = 1|a_1) - p(Y_j = 0|a_1)}, \tag{24}$$

$$B(i, j) = \frac{(2b - C_r)p(Y_i = 1|a_1)}{C_r p(Y_i = 1|a_2)} \prod_{k \neq i,j} \frac{p(\hat{Y}_k = 1|a_1)}{p(\hat{Y}_k = 1|a_2)}. \tag{25}$$

Proof: When (23) holds, it can be easily shown that $\boldsymbol{U^{D,2}_j}(\boldsymbol{p^c})$ is a continuous and strictly concave function of p^c_j (see Appendix B), for $j = 1, 2, \cdots, N$. In addition, the action set is closed, bounded and convex. By Theorem 3, the second-layer game admits an NE in pure strategy. ■

Note that the concavity of the utility function makes problem (13) a convex optimization problem, which is easy to solve numerically. Suppose that all the entities solve the corresponding convex optimization problems (13) asynchronously and broadcast their misreport probabilities according to their own timescale. Let T^j_u 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 update infinitely often), an asynchronous dynamic update algorithm is proposed to compute the NE of the second-layer game as in Algorithm 1.

Algorithm 1 Asynchronous Dynamic Update Algorithm

```
Initialization: set $t = 0$, $p^c_j = 0$ for $j = 1, 2, \cdots, N$
repeat
    for all $t = 0, 1, ..., N$ do
        if $t \in T^j_u$ then
            entity $j$ solves the convex optimization problem and updates $p^c_j(t)$.
        else
            $p^c_j(t) = p^c_j(t-1)$
        end if
    end for
    t=t+1
until converged
```

[7] Note that this condition always holds when the network is large enough, i.e., $N \to \infty$.

V. Byzantine Entities

The Byzantine attack, in which the malicious nodes deliberately send out falsified information to disrupt the normal information process, is first proposed in [25] and later extended to various distributed computing applications, including cooperative spectrum sensing (CSS) [26], distributed event detection [27], and distributed source coding [28], among the many others [29, 30]. Detecting the Byzantine attacks is challenging since the attackers can change its behavior arbitrarily. Not surprisingly, the Byzantine attack is also a potential threat to the privacy-aware collaborative system proposed in this work. In particular, the Byzantine attackers (e.g., a compromised entity) may share or inject tampered local detection results to other entities so as to devastate the entire collaboration team. In this section, the impact of the Byzantine attacks to the proposed framework and the corresponding solution are discussed.

For the discussion in the previous sections, it is assumed that all the collaborative entities will send their obfuscated observations and misreport probabilities honestly. Nonetheless, this assumption may not hold in practice. 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 and transformed to the Byzantine attackers), they may send out wrong observations and misreport probabilities deliberately, which will mislead the other entities. To represent a general setting, it is assumed in the following that at most f entities (except the defender) may be Byzantine faulty and may behave arbitrarily [25]. In addition, the external attacker can collude with the Byzantine faulty entities, and therefore knows their possible behaviors. Furthermore, a fully-distributed setting is considered, in which all the entities (including the defender) share their obfuscated observations and misreport probabilities with all the others. The objective of the entities is to collaboratively estimate the state of the network, and then choose the optimal responding strategy based on the estimation.

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

$$F^j(s=1|\hat{Y}_j) = \frac{p(\hat{Y}_j|s=1)p(s=1)}{p(\hat{Y}_j)}, \tag{26}$$

$$F^j(s=0|\hat{Y}_j) = 1 - F^j(s=1|\hat{Y}_j), \tag{27}$$

in which $p(s)$ is the distribution of the network state determined by the attacking strategy of the attacker (i.e., $p(s=1|a_1)=1, p(s=0|a_2)=1$); $p(\hat{Y}_j|s=1)$ is the distribution of the obfuscated observation $\hat{Y}_j$ given the network state $s=1$, which can be obtained by (4) and (5); $p(\hat{Y}_j)$ is the distribution of the obfuscated observation $\hat{Y}_j$ which is given by

$$p(\hat{Y}_j) = p(\hat{Y}_j|s=1)p(s=1) + p(\hat{Y}_j|s=0)p(s=0). \tag{28}$$

Therefore, after receiving the obfuscated observation and the misreport probability of entity j, all the entities can obtain a vector $\boldsymbol{w}_j = [w_j(0), w_j(1)] = [F^j(s=0|\hat{Y}_j), F^j(s=1|\hat{Y}_j)]$ denoting the log of the distribution of the network state from entity j's view after local processing. As a result, the problem can be formulated as a Byzantine vector consensus (BVC) problem [31, 32]. Considering that exact consensus is impossible in asynchronous systems in the presence of even a single crash failure [33], the Approximate BVC is considered, which must satisfy the following conditions:

•$\epsilon - Agreement$: The elements of the decision vectors at any two non-faulty entities must be within ϵ of each other, where $\epsilon > 0$ is a pre-defined constant.

•$Validity$: The decision vector at each non-faulty entity must be in the convex hull of the input vectors at the non-faulty entities.

•$Termination$: Each non-faulty entity must terminate within a finite amount of time.

To solve the Approximate BVC problem, the iterative algorithm in [31] is adopted, which is mainly based on the following result by Tverberg [34].

Theorem 4. *For any integer $f \geq 0$, and for every multiset Y containing at least $(d+1)f+1$ points in $\mathbb{R}^d$, there exists a partition $Y_1, \cdots, Y_{f+1}$ of Y into $f+1$ non-empty multisets such that the intersection of the convex hulls of the $f+1$ multisets is non-empty.*

The partition in Theorem 4 is called a Tverberg partition, and the points in the intersection of the convex hulls of the multisets are called Tverberg points. The algorithm is summarized as in Algorithm 2. According to [31], Algorithm 2 is guaranteed to converge to the consensus point when $N \geq (d+2)f+1$. Interested readers may refer to [31, 32] for more details.

To this end, we classify the Byzantine faulty entities into two types: The first type of faulty entities share incorrect obfuscated observations and misreport probabilities but follow the approximate BVC algorithm. Therefore, their final decision vectors will be the same as those of the non-faulty entities. The second type of faulty entities do not follow the approximate algorithm, and their final decision vectors will be different from the non-faulty entities. Therefore, these faulty entities can be identified by the defender and their shared information can be discarded. Based on the Byzantine Vector Consensus obtained in Algorithm 2, the defender further estimates the probability of the attacker launching an attack and chooses the optimal responding strategy. The corresponding algorithm is summarized as in Algorithm 3.

Theorem 5. *Let $\mathcal{N}_g$ denote the set of non-faulty entities, when $N \geq (d+2)f+1$ and $F^i(s|\hat{Y}_i) = F^j(s|\hat{Y}_j), \forall i,j \in \mathcal{N}_g, s \in \{0,1\}$, the only possible decision vector of all non-faulty entities running Algorithm 2 is identical to their input vectors.*

Proof: According to the $Validity$ condition, the decision vector at each non-faulty entity must be in the convex hull of

Algorithm 2 Approximate BVC Algorithm with input $\boldsymbol{v}_j[0] = \log(\boldsymbol{w}_j)$ at entity j

```
1. Initialization step: Each entity obtains the initial distribution of the network state $\boldsymbol{v}_j[0]$. In each round (indexed by $k$), the update process is divided into three steps: transmit step, receiving step and update step.
2. Transmit step: Each entity $j$ broadcasts its current state distribution vector $\boldsymbol{v}_j[k-1]$ to all the other entities.
3. Receiving step: Each entity $j$ receives the shared vectors $\boldsymbol{v}_j[k-1], \forall j$. These vectors form a multiset $r_j[k]$.
4. Update step: Form a multiset $Z_j[k]$ using the steps below:
5.     • Initialize $Z_j[k]$ as empty.
6.     • Add to $Z_j[k]$ any one Tverberg point corresponding to each multiset $C \subseteq r_j[k]$ such that $|C| = (d+1)f+1, d = 2$. By Theorem 4, such a Tverberg point exists.
7.     • Compute new state distribution vector as
           $\boldsymbol{v}_j[k] = \frac{\boldsymbol{v}_j[k-1] + \sum_{\boldsymbol{z} \in Z_j[k]} \boldsymbol{z}}{1 + |Z_j[k]|}$   (29)
8.     • Normalize $\boldsymbol{v}_j[k]$.
9. Termination: Each non-faulty entity terminates after completing $t_{end}$ iterations, where $t_{end}$ is a pre-defined constant chosen according to [32].
```

Algorithm 3 Approximate BVC based Algorithm for Collaborative Security

```
1. Initialization step: All entities observe the network state, obfuscate the observations after they determine the misreport probabilities, and estimate the distribution of the network state, set as $\boldsymbol{v}_j[0] = [\log(F^j(s=0|\hat{Y}_j)), \log(F^j(s=1|\hat{Y}_j))], \forall j$.
2. Run Algorithm 2 and obtain the decision vector at Byzantine Vector Consensus $\hat{\boldsymbol{v}}_j = [\log(\hat{F}^j(s=0|\hat{Y}_j)), \log(\hat{F}^j(s=1|\hat{Y}_j))], \forall j$.
3. The defender identifies the entities whose decision vectors are different from the decision vector at BVC as faulty entities and discards the information from them.
4. The defender computes the new $\frac{p(\hat{Y}_j|s=0)}{p(\hat{Y}_j|s=1)}, \forall j \in \mathcal{N}_g$ using the equation below:
       $\frac{p(\hat{Y}_j|s=0)}{p(\hat{Y}_j|s=1)} = \frac{p(s=1)}{p(s=0)}\Big[\frac{1}{\hat{F}^j(s=1|\hat{Y}_j)} - 1\Big],$
   in which $\mathcal{N}_g$ is the set of non-faulty entities identified previously.
5. The defender estimates the probability of the attacker launching an attack (i.e., $F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{-\boldsymbol{i}})$) using the equation below:
       $F^i(a_1|Y_i, \hat{\boldsymbol{Y}}_{-\boldsymbol{i}}) = \frac{1}{1 + \frac{p(s=0)}{p(s=1)}\frac{p(Y_i|s=0)}{p(Y_i|s=1)}\prod_{j\neq i, j\in\mathcal{N}_g}\frac{p(\hat{Y}_j|s=0)}{p(\hat{Y}_j|s=1)}}.$
6. The defender determines the optimal responding strategy accordingly.
```

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 3. *When $N \geq (d+2)f+1$ and $F^i(s|\hat{Y}_i = 1) = F^j(s|\hat{Y}_j = 1), \forall i, j \in \mathcal{N}_g, s \in \{0,1\}$, 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

$$p_*^A(a_1) = \frac{C_r}{(2b - C_r)\frac{p(Y_i=1,\hat{\mathbf{Y}}_{-\boldsymbol{i}}=\mathbf{1}|a_1)}{p(Y_i=1,\hat{\mathbf{Y}}_{-\boldsymbol{i}}=\mathbf{1}|a_2)} + C_r}$$
$$= \frac{C_r}{[(2b - C_r)\frac{p(Y_i=1|s=1)}{p(Y_i=1|s=0)}\prod_{j\neq i, j\in\mathcal{N}_g}\frac{p(\hat{Y}_j=1|s=1)}{p(\hat{Y}_j=1|s=0)}] + C_r}. \tag{30}$$

According to Theorem 5, when $N \geq (d+2)f+1$ and $F^i(s|\hat{Y}_i = 1) = F^j(s|\hat{Y}_j = 1), \forall i, j \in \mathcal{N}_g, s \in \{0,1\}$, 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 for the non-faulty entities, $\frac{p(\hat{Y}_j=1|s=1)}{p(\hat{Y}_j=1|s=0)}$ remains the same for all $j \in \mathcal{N}_g$. For the faulty entities that do not follow the approximate BVC algorithm, their decision vectors will be different from those of the non-faulty entities, and thus will be identified and the information shared by them can be discarded by the defender. For the faulty entities that follow the approximate BVC algorithm, their decision vectors will be the same as those of the non-faulty entities. Therefore, there will be at least $N-f$ entities been identified as non-faulty (i.e., $|\mathcal{N}_g| \geq N-f$). In this case, the expected payoffs of the entities will be no less 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. ■

Remark 3. *In the case that the non-faulty entities have the same observation capabilities and privacy requirements, the conditions in Corollary 3 will be satisfied and the proposed algorithm can guarantee a desired collaboration gain. In other cases, the performance of the proposed algorithm depends on the specific settings.*

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 entities, and it is assumed that all of them have high security requirements, with powerful response capability and insignificant cost of response (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 $j = 1, 2, \cdots, N$.[8]

[8] These parameters are chosen mainly for the illustration purpose. In practice, these parameters can be set according to relevant applications.

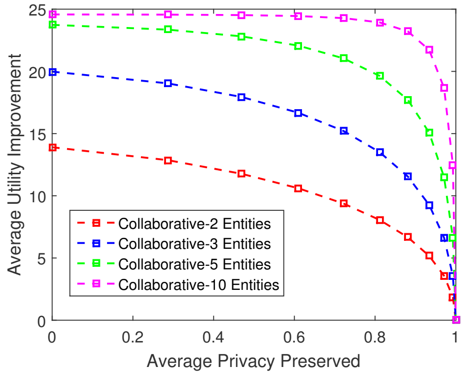

Fig. 2. Utility-Privacy tradeoff curve.

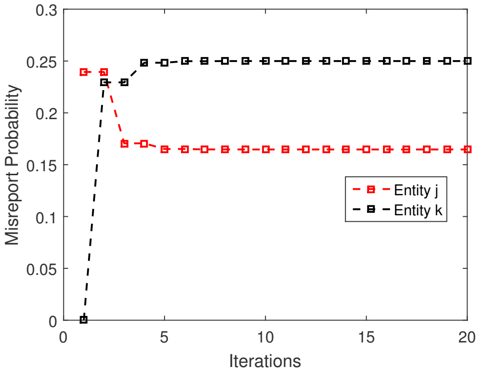

Fig. 3. Convergence of the Asynchronous Dynamic Update Algorithm.

TABLE II
MISREPORT PROBABILITY WITH RESPECT TO λ_j

λ_j	0	5	10	15	20
p_j^c	0	10%	23%	30%	34%
p_k^c	30%	27%	23%	21%	20%

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 the defender) and the preserved privacy (i.e., $1 - P_L(p_j^c)$) of all the entities. It can be seen that in all the examined scenarios, the collaborative scheme always enhances the performance, which justifies Corollary 2. In addition, the payoff improvement achieves its highest value when the preserved privacy is 0 (i.e., all the entities share their observations with the defender 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 preserve more privacy while achieving the same the payoff improvement, or equivalently achieve higher collaboration gain for a given privacy preservation requirement. Intuitively, when there are more collaborative entities, the defender can gather more information about whether an attack is launched 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 the three collaborative entities case are examined (i.e., entity j and entity k share their observations with the defender). Similar results are observed for the cases of $N > 3$.

Assuming $\lambda_k = 10$, Table. II shows how λ_j will influence the collaboration strategies of both entities. In our model, λ_j determines how important privacy is for entity j, and is thus closely related to its privacy requirement (c.f. (9)). It can be seen that with different λ_j, not only the misreport probability of entity j changes but also that of entity k. More specifically, when λ_j becomes larger, entity j will collaborate with higher misreport probability while entity k will collaborate with lower misreport probability. This may be explained as follows: a larger λ_j implies that entity j emphasizes more privacy, and hence it will prefer to increase its misreport probability. In the meantime, with larger p_j^c, the second-layer game transits to another NE point. Recall that the payoff functions of the entities (i.e., $\boldsymbol{U_*^D}(\boldsymbol{p^c})$) are concave functions of both p_j^c and p_k^c. As p_j^c becomes larger, the payoff improvement brought by decreasing misreport probability more than compensates the corresponding privacy loss for entity k, which further encourages it to collaborate will lower misreport probability.

In addition, it is worth mentioning that for different privacy requirements (i.e., different λ_j and λ_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. With the same setting as in Section VI-B and assuming that $\lambda_j = 7, \lambda_k = 10$, Fig. 3 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 λ_j and λ_k), which verifies the effectiveness of Algorithm 1.

D. Byzantine Entities

In this subsection, the impact of Byzantine entities is examined. A network that consists of 5 collaborative 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 λ) and thus will misreport with the same probability. Fig. 4 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

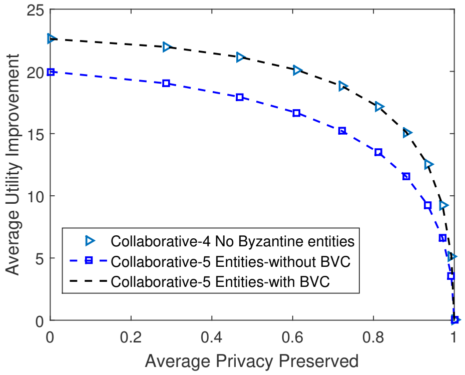

Fig. 4. Influence of Byzantine entities.

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 as indicated by Corollary 3.

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, significant research efforts have been devoted to the security-related information sharing framework, and various game-theoretic methods have been developed. For example, [4] proposed a two-player game model to help understand the benefit of information sharing and pointed out how the characteristics of the entities affect the incentives for information sharing. In [5] and [6], a two-player game between two competing firms which share a common platform was formulated. By game-theoretic analysis, the tradeoff between security investment and breach information sharing was studied and discussed. [7] used a two-stage Bayesian game to analyze the information sharing decisions of the two competing firms. [8] modeled the information exchange among the firms as a distributed non-cooperative game and found the best investment and sharing strategies. [9] considered a set of users in a public cloud who share the same hypervisors and obtained the necessary conditions under which a rational user in a public cloud will share his discovered vulnerabilities by analyzing the NE strategies of the proposed two-player game. However, these work fail to consider the influence of the attacker and the privacy issues induced by information sharing.

Considering the potential privacy leakage in information sharing, some privacy-aware collaborative security schemes have been developed. In [11] and [12], privacy of sensitive data from the distributed alerts could be partially preserved by the utilization of Bloom filters due to their probabilistic data structure. In [13] and [16] cryptographic methods were used to preserve the sensitive information in intrusion alert data sharing. [14] and [15] proposed the use of entropy guided alert sanitization, where sensitive attributes of the alerts were generalized to high-level concepts to introduce uncertainty into the dataset. Among all these privacy-aware collaborative security schemes, none of them was able to quantitatively study the tradeoff between the utility and privacy. Different from the works mentioned above, our work studies the utility-privacy tradeoff from a game-theoretic viewpoint, and derives the NE and SNE which provide the optimal collaboration as well as response strategies of the collaborative entities for different privacy requirements.

VIII. Conclusions and Future Works

In this work, the utility-privacy tradeoff problem in collaborative 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 impact of Byzantine entities is also investigated. Further extending this work to dynamic settings and multiple-attack settings constitute interesting future directions.

Appendix A
Proof of Theorem 2

Proof: According to (10),

$$\boldsymbol{U_t^A}(\boldsymbol{p_t^A}, \boldsymbol{p_t^D}(\boldsymbol{p_t^A})) = \sum_{Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}} \in \{0,1\}^N} \big[p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) p_t^A(a_1) p_t^D(d_2|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})(W - C_aW) + p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) p_t^A(a_1) p_t^D(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})[(1-2b-C_a)W]\big], \tag{31}$$

where $p_t^D(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}})$ is given by

$$p_t^D(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = \begin{cases} 1 & \text{if } F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) > \frac{C_r}{2b}, \\ \in [0,1] & \text{if } F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = \frac{C_r}{2b}, \\ 0 & \text{if } F^i(a_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) < \frac{C_r}{2b}. \end{cases} \tag{32}$$

Let

$$\sum_{Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}} \in \{0,1\}^N} p(Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) p_t^D(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = p_t^{Res}, \tag{33}$$

we have

$$\boldsymbol{U_t^A}(\boldsymbol{p_t^A}, \boldsymbol{p_t^D}(\boldsymbol{p_t^A})) = p_t^A(a_1)(1 - C_a - 2bp_t^{Res})W. \tag{34}$$

The attacker finds its optimal strategy by solving the following optimization problem (c.f. (9)):

$$\boldsymbol{p_t^A}(\boldsymbol{p_t^D}) = \operatorname*{argmax}_{\boldsymbol{p_t^A}} \sum_{t=1}^{T_e} \boldsymbol{U_t^A}(\boldsymbol{p_t^A}, \boldsymbol{p_t^D}(\boldsymbol{p_t^A})). \tag{35}$$

A. *Case 1*

If the attacker chooses its attacking strategies $p_t^A(a_1) = p_t^L$ such that $p_t^D(d_1|Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) = 0, \forall (Y_{i,t}, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}}) \in \{0,1\}^N (i.e., p_t^{Res} = 0), \forall t$, which means the best response of the defender is to choose "do nothing", the expected total payoff of the attacker (denoted by $U(0)$) is given by

$$U(0) = \sum_{t=1}^{T_e} \boldsymbol{U_t^A}(\boldsymbol{p_t^A}, \boldsymbol{p_t^D}(\boldsymbol{p_t^A})) = T_e p_t^A(a_1)[1 - C_a]W, \tag{36}$$

where T_e is infinite as the defender never responds.

B. *Case 2*

If the strategies chosen by the attacker are as follows

$$p_t^A(a_1) = \begin{cases} p_t^L & \text{if } t \notin T_K, \\ p_t^H & \text{if } t \in T_K. \end{cases}$$

such that

$$p_t^{Res} = \begin{cases} 0, & \text{if } t \notin T_K, \\ \in (0,1] & \text{if } t \in T_K, \end{cases}$$

where $T_K = \{t_1, \cdots, t_K\}$ is the set of time instants when the attacker chooses its attacking strategies p_t^H such that $p_t^{Res} > 0, \forall t \in T_K$ and $t_m < t_n, \forall m < n$, it can be shown that the expected payoff of the attacker is always smaller than that of the Case 1.

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

$$U(1) = \sum_{t=1}^{T_e} \boldsymbol{U_t^A}(\boldsymbol{p_t^A}, \boldsymbol{p_t^D}(\boldsymbol{p_t^A})) = G(t_1) + \sum_{t=1}^{t_1-1} [p_t^L(1 - C_a)W] \\ + [1 - bp_{t_1}^H p_{t_1}^{Res}] \sum_{t=t_1+1}^{T_e} p_t^L(1 - C_a)W, \tag{37}$$

where $bp_{t_1}^H p_{t_1}^{Res}$ is the probability that the attacker launches an attack and the defender responds to the attack successfully 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

$$U(0) - U(1) = bp_{t_1}^H p_{t_1}^{Res} \sum_{t=t_1+1}^{T_e} p_t^L[1 - C_a]W \\ + p_t^L(1 - C_a)W - G(t_1). \tag{38}$$

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

2) $K > 1$: Assuming that we now have $K = k$ where $k \geqslant 1$, if the attacker further chooses to launch an attack with probability $p_{t_{k+1}}^H$ such that $p_{t_{k+1}}^{Res} > 0$ 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 $t \geqslant t_{k+1}$ will decrease according to the discussion above. As a result, $U(k) > U(k+1), \forall k \geqslant 1$. Therefore

$$U(0) > U(k), \forall k > 0, \tag{39}$$

which means the optimal strategy of the attacker is to choose $p_t^A(a_1) = p_t^L$ such that $p_t^{Res} = 0, \forall t$. By (34), when $p_t^{Res} = 0, \forall t$, the attacker's utility function is an increasing function of $p_t^A(a_1)$, therefore, the optimal strategy of the attacker is $p_t^A(a_1) = p_*^A(a_1)$, which corresponds to $F^i(a_1|Y_{i,t} = 1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i,t}} = \boldsymbol{1}) = \frac{C_r}{2b}$. ■

Appendix B
Concavity of second-layer game utility function

Proof: According to the discussion in Section IV,

$$\boldsymbol{U_*^D}(p_j^c) = \frac{-C_r W}{(2b - C_r)\frac{p(Y_i=1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_1)}{p(Y_i=1, \hat{\boldsymbol{Y}}_{\boldsymbol{-i}}=\boldsymbol{1}|a_2)} + C_r}$$
$$= \frac{-C_r W}{[(2b - C_r)\frac{p(Y_i=1|a_1)}{p(Y_i=1|a_2)} \prod_{k \neq i,j} \frac{p(\hat{Y}_k=1|a_1)}{p(\hat{Y}_k=1|a_2)}] \frac{p(\hat{Y}_j=1|a_1)}{p(\hat{Y}_j=1|a_2)} + C_r},$$

where

$$\frac{p(\hat{Y}_j = 1|a_1)}{p(\hat{Y}_j = 1|a_2)} = \frac{p(Y_j = 1|a_1)(1 - p_j^c) + p(Y_j = 0|a_1)p_j^c}{p(Y_j = 1|a_2)(1 - p_j^c) + p(Y_j = 0|a_2)p_j^c}. \tag{40}$$

Let

$$a = C_r p(Y_j = 1|a_2), \tag{41}$$

$$b = C_r[p(Y_j = 0|a_2) - p(Y_j = 1|a_2)], \tag{42}$$

$$m = [(2b - C_r)\frac{p(Y_i = 1|a_1)}{p(Y_i = 1|a_2)} \prod_{k \neq i,j} \frac{p(\hat{Y}_k = 1|a_1)}{p(\hat{Y}_k = 1|a_2)}]\times \\ p(Y_j = 1|a_1) + C_r p(Y_j = 1|a_2), \tag{43}$$

$$n = [(2b - C_r)\frac{p(Y_i = 1|a_1)}{p(Y_i = 1|a_2)} \prod_{k \neq i,j} \frac{p(\hat{Y}_k = 1|a_1)}{p(\hat{Y}_k = 1|a_2)}]\times \\ [p(Y_j = 0|a_1) - p(Y_j = 1|a_1)] + b. \tag{44}$$

Then, (40) can be expressed as

$$\boldsymbol{U_*^D}(p_j^c) = -\frac{a + bp_j^c}{m + np_j^c} W. \tag{45}$$

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 $\boldsymbol{U_*^D}(p_j^c)$ is determined by n.

A. *Case 1: $n \neq 0$*

In this case, (45) can be expressed as

$$\boldsymbol{U_*^D}(p_j^c) = -\left[\frac{b}{n} + \frac{\frac{a}{n} - \frac{bm}{n^2}}{\frac{m}{n} + p_j^c}\right] W = -\left[\frac{b}{n} + \frac{1}{n^2}\frac{an - bm}{\frac{m}{n} + p_j^c}\right] W, \tag{46}$$

where $an - bm < 0$. Therefore, the sufficient and necessary condition for $\boldsymbol{U_*^D}(p_j^c)$ being strictly concave when $p_j^c \in [c_j, 0.5]$ is given by

$$\frac{m}{n} < -\frac{1}{2}, \tag{47}$$

which is equivalent to

$$A(j) < B(i,j), \tag{48}$$

where

$$A(j) = \frac{p(Y_j = 0|a_2) - p(Y_j = 1|a_2)}{p(Y_j = 1|a_1) - p(Y_j = 0|a_1)} \tag{49}$$

$$B(i,j) = \frac{(2b - C_r)p(Y_i = 1|a_1)}{C_r p(Y_i = 1|a_2)} \prod_{k \neq i,j} \frac{p(\hat{Y}_k = 1|a_1)}{p(\hat{Y}_k = 1|a_2)}. \tag{50}$$

Note that given fixed $\boldsymbol{p^c_{-j}}$, $\boldsymbol{U^D_{j,*}}(\boldsymbol{p^c_{-j}}, p^c_j = 0.5)$ is constant. Furthermore, when $\lambda_j \geq 0$, $-\lambda_j P_L(p^c_j)$ is also concave by its definition in (12). Therefore, $\boldsymbol{U^{D,2}_j}(\boldsymbol{p^c})$ is a continuous and strictly concave function of p^c_j when (48) holds.

B. Case 2: $n = 0$

When $n = 0$, we have $A(j) = B(i,j)$. In this case, $\boldsymbol{U^D_*}(p^c_j) = -\frac{a+bp^c_j}{m}W$. Apparently, it is concave but not strictly concave in terms of p^c_j. In addition, when $\lambda_j > 0$, $-\lambda_j P_L(p^c_j)$ is strictly concave. Therefore, $\boldsymbol{U^{D,2}_j}(\boldsymbol{p^c})$ is a continuous and strictly concave function of p^c_j when $A(j) = B(i,j)$ and $\lambda_j > 0$. ■

References

[1] J. M. Seigneur, *Collaborative Computer Security and Trust Management.* IGI Global, 2009.

[2] G. Meng, Y. Liu, J. Zhang, A. Pokluda, and R. Boutaba, "Collaborative security: A survey and taxonomy," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1:1–1:42, Jul. 2015.

[3] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 55:1–55:33, May 2015.

[4] E. Gal-Or and A. Ghose, "The economic consequences of sharing security information," in *Economics of information security.* Springer, 2004, pp. 95–104.

[5] L. A. Gordon, M. P. Loeb, and W. Lucyshyn, "Sharing information on computer systems security: An economic analysis," *Journal of Accounting and Public Policy*, vol. 22, no. 6, pp. 461–485, 2003.

[6] D. Liu, Y. Ji, and v. Mookerjee, "Knowledge sharing and investment decisions in information security," *Decision Support Systems*, vol. 52, no. 1, pp. 95–107, 2011.

[7] M. H. R. Khouzani, V. Pham, and C. Cid, "Strategic discovery and sharing of vulnerabilities in competitive environments," in *International Conference on Decision and Game Theory for Security.* Springer, 2014, pp. 59–78.

[8] D. K. Tosh, S. Sengupta, S. Mukhopadhyay, C. A. Kamhoua, and K. A. Kwiat, "Game theoretic modeling to enforce security information sharing among firms," in *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on.* IEEE, 2015, pp. 7–12.

[9] C. Kamhoua, A. Martin, D. K. Tosh, K. A. Kwiat, C. Heitzenrater, and S. Sengupta, "Cyber-threats information sharing in cloud computing: A game theoretic approach," in *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on.* IEEE, 2015, pp. 382–389.

[10] P. Gross, J. Parekh, and G. Kaiser, "Secure selecticast for collaborative intrusion detection systems," in *Proceedings of the 3rd International Workshop on Distributed Event-Based Systems (DEBS04).* IET, 2004.

[11] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Towards collaborative security and p2p intrusion detection," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, June 2005, pp. 333–339.

[12] E. Vasilomanolakis, M. Krügl, C. G. Cordero, M. Mühlhäuser, and M. Fischer, "Skipmon: A locality-aware collaborative intrusion detection system," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, Dec 2015, pp. 1–8.

[13] P. Lincoln, P. A. Porras, and V. Shmatikov, "Privacy-preserving sharing and correlation of security alerts," in *USENIX Security Symposium*, 2004, pp. 239–254.

[14] D. Xu and P. Ning, "Privacy-preserving alert correlation: a concept hierarchy based approach," in *21st Annual Computer Security Applications Conference (ACSAC'05)*, Dec 2005.

[15] ——, "A flexible approach to intrusion alert anonymization and correlation," in *Securecomm and Workshops, 2006*, Aug 2006, pp. 1–10.

[16] H. G. Do and W. K. Ng, "Privacy-preserving approach for sharing and processing intrusion alert data," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, April 2015, pp. 1–6.

[17] J. Cheng, S. H. Wong, H. Yang, and S. Lu, "Smartsiren: virus detection and alert for smartphones," in *Proceedings of the 5th international conference on Mobile systems, applications and services.* ACM, 2007, pp. 258–271.

[18] J. Reed, A. J. Aviv, D. Wagner, A. Haeberlen, B. C. Pierce, and J. M. Smith, "Differential privacy for collaborative security," in *Proceedings of the Third European Workshop on System Security.* ACM, 2010, pp. 1–7.

[19] E. T. Jaynes, *Information theory and statistical mechanics.* Physical review 106.4: 620, 1957.

[20] L. Chen and J. Leneutre, "A game theoretical framework on intrusion detection in heterogeneous networks," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 2, pp. 165–178, June 2009.

[21] M. J. Osborne and A. Rubinstein, *A Course in Game Theory.* Cambridge, MA: MIT Press, 1994.

[22] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos, "Security decision-making among interdependent organizations," in *2008 21st IEEE Computer Security Foundations Symposium*, June 2008, pp. 66–80.

[23] Q. Zhu, C. Fung, R. Boutaba, and T. Basar, "A game-theoretical approach to incentive design in collaborative intrusion detection networks," in *Proc. International Symp. Game Theory Netw.*, Istanbul, May. 2009.

[24] T. Basar and G. J. Olsder, *Dynamix Noncooperative Game Theory.* 2nd ed. SIAM, Philadelphia, 1998.

[25] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[26] X. He, H. Dai, and P. Ning, "A Byzantine attack defender: The conditional frequency check," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on.* IEEE, 2012, pp. 975–979.

[27] P. Zhang, J. Y. Koh, S. Lin, and I. Nevat, "Distributed event detection under Byzantine attack in wireless sensor networks," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on.* IEEE, 2014, pp. 1–6.

[28] O. Kosut and L. Tong, "Distributed source coding in the presence of byzantine sensors," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2550–2565, 2008.

[29] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 16–29, 2009.

[30] P. Goyal, S. Batra, and A. Singh, "A literature review of security attack in mobile ad-hoc networks," *International Journal of Computer Applications*, vol. 9, no. 12, pp. 11–15, 2010.

[31] N. H. Vaidya and V. K. Garg, "Byzantine vector consensus in complete graphs," in *Proceedings of the 2013 ACM symposium on Principles of distributed computing.* ACM, 2013, pp. 65–73.

[32] N. H. Vaidya, "Iterative byzantine vector consensus in incomplete graphs," in *International Conference on Distributed Computing and Networking.* Springer, 2014, pp. 14–28.

[33] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985.

[34] M. A. Perles and M. Sigron, "A generalization of tverberg's theorem," *arXiv preprint arXiv:0710.4668*, 2007.