

Information concealing games

Saswati Sarkar, Eitan Altman, and Pramod Vaidyanathan

Abstract—A system with an n -dimensional state vector and a *controller* and an *actor* is considered. The controller has complete information about the system state, and reveals a certain “minimum” amount of information to the actor. The actor takes certain actions based on the information the controller reveals, and the actions fetch certain utilities for each entity. Both the controller and actor seek to maximize their individual utilities by respectively selecting the information to reveal and the actions to adopt. This decision problem forms the basis of several technical and social systems, and can be formulated as a signaling game. It is shown that the Perfect Bayesian Equilibrium of this game has several counter-intuitive properties and can be obtained as a saddle point of a different two person zero sum game. The computation time for saddle points using standard linear programs however turns out to be super-exponential in n , which leads to computational intractability even for moderate n . Algorithms for computing saddle point policies using a computation time that is exponential in n are presented. Finally, simple linear time computable policies that approximate the saddle-point policies within guaranteeable approximation ratios are obtained.

Keywords: approximation algorithms, computational tractability, information concealing, leader follower games, perfect Bayesian equilibrium, signaling games

I. INTRODUCTION

A. Overview

Exchange of information among different entities forms the basis of most technological advances in the information era and also of social interactions. Several seminal advances in communication systems have led to schemes that maximize the rate of exchange of information. An aspect that has received somewhat less attention, and is as important, is that of designing a framework for deciding what information should be revealed and what should be concealed during exchange of information among different entities so as to maximize their utilities. The main challenge towards developing such a framework is that oftentimes such decisions depend on the objective for exchange of information, and hence can only be determined on a case by case basis. The contribution of this paper is to develop a rigorous mathematical framework for deciding what information an entity should reveal when the objectives satisfy certain broad characterizations that capture the essence of several communication and social systems.

We consider a system with two entities. The state of the system is a random vector of dimension n . At any given time the first entity (*controller*) has complete information about the state of the system, and must reveal a certain “minimum” amount of information about the system state to the second entity. It can however choose the nature of the information it reveals subject to satisfying the above constraint. The second entity (*actor*) takes certain actions based on the information the controller reveals, and the actions are associated with certain utilities for both the controller and the actor which also depend on the state of the system. The same actions and the

system states fetch different utilities for the controller and the actor, and usually when one entity has a high utility the other has a low utility. We devise a framework that enables the controller to decide the information it would reveal, or equivalently conceal, so as to maximize its own utility, and the actor to determine its actions based on the information the controller reveals so as to again maximize its utility.

B. Motivation

We first establish that this information concealing problem forms the basis of several communication systems.

1) Information concealing problems in wireless networks:

a) Cognitive radio networks: Consider a transmitter with access to n channels, whose qualities constitute the state of the system. The transmitter needs to select one channel for transmission, and the transmission quality of the selected channel determines the rate of successful transmission. Hence, the transmitter probes the channels in order to assess their qualities before it transmits any packet. A malicious entity, say a jammer, seeks to reduce the rate of successful transmission. The jammer is usually assumed to accomplish its goal by generating signals that interfere with the transmitter’s communication; however the jammer may be able to deteriorate the transmission rate much more by preventing the transmitter from learning the states of the channels. This may cause the transmitter to make a wrong choice, that is, select a channel with a poor transmission quality, and thereby obtain a poor data rate for a while. Note that the jammer can prevent the transmitter from learning the states of some channels, possibly by generating signals that interfere with the corresponding probe packets or responses to these probes, and generating such signals may consume less energy as compared to those that jam the actual transmission since the probe packets are transmitted over shorter durations. We therefore consider the case where the jammer blocks the probe packets and not the actual transmission. Furthermore, we assume that the jammer knows the quality of the channels and can block the probes in at most k channels since the blocking process consumes energy. Hence, the states of at most k channels can be concealed from the transmitter. The transmitter selects the channel after it learns about the states of the channels the jammer does not conceal. Note that the transmitter may either select a channel whose state has been revealed or one whose state has been concealed; the latter may happen since the fact that the jammer has concealed the state of a channel may indicate that the transmission quality of the corresponding channel is good. The rate of successful transmission attained by the transmitter determines the utility of the transmitter and the jammer. The information concealing problem we described will enable the jammer (controller) to optimally determine which channels it would conceal, and the transmitter (actor) to select the channel.

2) Information concealing problems in other information systems:

a) Query resolution networks: We next describe another communication system in which the information concealing problem arises. Consider a client that needs to locate a desired information. It queries some data bases to determine which of them has the information. The responses constitute the state of the system and

S. Sarkar is with the Dept. of Electrical and Systems Eng., University of Pennsylvania 200 S. 33rd Street, Philadelphia 19104, USA. E. Altman is at INRIA, 2004 Route des Lucioles, B.P.93, 06902 Sophia-Antipolis Cedex, France. P. Vaidyanathan is with Citigroup, 388 Greenwich Street, New York, NY 10013. Their emails are swati@seas.upenn.edu, Eitan.Altman@sophia.inria.fr and Pramod.Vaidyanathan@gmail.com. This work has been partially supported by the European Bionets project and NSF grants CNS-0915697, CNS-0915203, CNS-0914955, CNS-0721308.

Parts of this paper have been presented at Infocom mini-conference [11].

specify the probability with which the requested information is present in the data base (as the search in response to such preliminary queries may not be comprehensive and also the information may be dated). The responses reach the node through a gateway that has a malicious entity which blocks some of the responses in order to undermine the information location service. The client needs to determine which database it would request the information from based on the responses to its query, and again it may choose one it received a response from or one it did not receive a response from (the latter may happen if the responses it receives reveal low probabilities). The utility of the client and the malicious entity depends on the probability that the client obtains the information it is interested in. The information concealing problem we described will enable the malicious entity (controller) to optimally determine which responses it would suppress and the client (actor) to select the database it would request the information from.

b) Buyer-Seller authentication in e-commerce: Consider an e-commerce system where a buyer and a seller are bargaining. The authentication process between them proceeds in two stages. The buyer has n pieces of information using which he can authenticate himself to the seller. He reveals limited information about k of these pieces using which the seller can complete the first stage of the authentication successfully if the buyer is who he claims to be (e.g., using some proof verification methods). Next, the seller identifies himself to the buyer, and subsequently asks about complete information for one of the n pieces which may or may not be among those that the buyer initially selects. The buyer provides the requested information and the authentication is successful if again he is who he claims to be. This two-stage authentication process allows each entity to identify himself once he has some (albeit incomplete) information about the other participant. Now, the complete information the buyer reveals about any one piece in the authenticating process may allow the seller to acquire more information about the buyer than that required for mere authentication, e.g., information about his previous transactions with other merchants, etc. This will for example allow him to bargain more effectively with the buyer once the authentication is successful. Now, the different pieces of information the buyer possesses about himself reveals different amount of information about him, and the buyer must select the k pieces in the first stage so as to minimize the additional information he finally reveals to the seller. The seller must subsequently select the piece in the second stage to acquire maximum possible information about the buyer. The information concealing problem we described will enable the buyer (controller) and the seller (actor) to attain their respective objectives by optimally selecting the pieces in question.

3) Information concealing problems in social context:

a) Gambling: Consider a gambling game in which two gamblers have a common collection of N cards each of which can have one of m colors. They randomly select a number for each card and write the chosen number on one side of the corresponding card. Subsequently, the second gambler draws n cards randomly from the collection without observing the numbers on them. The first gambler then observes the colors and the numbers of the cards drawn and tells the second the numbers and the colors of k of these cards, and only the colors of the rest of the cards. The second gambler needs to select one of these n cards (either a card whose number it knows or one whose number it does not know), and the first pays him an amount that equals the number on the selected card (if this number is negative then the second pays the first). The first gambler (the controller) needs to select the k cards so as to minimize the amount it pays, and the second needs to select a card so as to maximize the amount it receives.

b) Security systems: Consider a corrupt employee who sells secrets about the company's security system to a burglar. The building in which the company is located has n gates, and the employee knows the efficacy of the security system at each gate (e.g., he may know the number of guards at each gate which may be a random variable owing to the company's security plan), and based on the price the burglar has offered or in order to conceal his collusion in the event of an enquiry, the employee informs the burglar about the security system of only $n - k$ of these gates. He also decides to select the gates whose information he reveals so as to minimize the probability that the break in is successful since if there is a successful break-in a comprehensive enquiry is likely to be launched. Based on the information he obtains from the employee, the burglar selects one gate through which he tries to enter; he selects this gate so as to maximize his probability of success.

In both these examples, the information concealing problem we described will enable the controller (first gambler or employee) and the actor (second gambler or burglar) to attain their objectives by making appropriate selections.

C. Contribution and Challenges

Our first contribution is to provide a framework for investigating information concealing problems. We formulate this problem as a signaling game ([5], Chapter 8.2) between two players and consider perfect Bayesian equilibrium (PBE) ([5], Chapter 8.2) as the solution concept (Section III). The subsequent challenge is to compute desirable equilibrium policies of the players as in general in signaling games multiple policy pairs attain this equilibrium and different equilibrium policies fetch different utilities for the same player. Also, general purpose algorithms for computing a PBE policy-pair are not known for arbitrary signaling games. We show that in the information concealing game all PBE policy-pairs fetch the same expected utility for each player - thus all such policy-pairs are functionally equivalent, and hence choice among them is not critical. We also show that PBE policy-pairs in this game can be computed by solving linear programs with finite number of variables and constraints. We prove the above by showing that there is a one-to-one correspondence between the set of PBE in the above game and the set of saddle points in a two-person zero-sum game (Section IV-A), which we refer to as an equivalent game. This equivalence holds although the original game is not a zero-sum game, and is an interesting result in itself as such equivalences are not commonly encountered between signaling games and two-person zero-sum games. Using this equivalence, we further demonstrate that several intuitively appealing policies of the controller and the actor do not in general constitute PBE. For example, in cognitive radio networks, a naive policy for the jammer that conceals the states of the channels that have the k best transmission qualities does not constitute a PBE. This happens because the actor can learn about the system not only from the information the controller reveals, but also from the choices of the controller.

We next investigate the computational tractability of the information concealing games. Our results in this area constitute our second contribution since general results that can address the computational aspects in this case are not known in the game theory or approximation algorithm literature. Note that the saddle-points of the equivalent games can be computed by solving standard linear programs (Chapter III.2.4, [6]), which would therefore provide a PBE of the original game as well. But, the number of variables and constraints in the standard linear program formulations for the equivalent games are super-exponential in n ($\Omega(e^{n2^n})$), where n is the dimension of the state-space of the system. Thus, the standard

linear programs become computationally intractable even for small values of n . Exploiting specific characteristics of the game under consideration, we next obtain linear programs which compute the saddle points of the equivalent game and the optimal policies for the two players while using exponential number of variables and constraints (Section IV-B). This significant reduction in computation time enables the computation of the optimal policies for moderate n . We next obtain linear time ($O(n)$) computable policies with provable performance guarantees for the two players (Section V). Specifically, these policies attain utilities that differ from the utilities of the saddle points by (a) constant factors in several important special cases, and (b) by factors that depend only on the amount of information that the controller reveals to the actor, and do not depend on n in the most general case. We also show that there exists examples where these performance guarantees are tight, which in turn allows us to complete characterize the performance of these policies.

II. RELATED LITERATURE

To the best of our knowledge, the information concealing game has not been investigated before. Information concealing game is however a special case of the well-known signaling games ([5], Chapter 8.2), and arises when the utilities of the two players in signaling games satisfy certain structure. The investigation of this special case has been motivated by its relevance in modeling a diverse range of applications in technical and social context, and also because a framework for computing the solutions and investigating their characteristics is not known for signaling games in general.

A game that is close to the information concealing games and has been investigated before is that introduced by noble-laureate P. Aumann *et. al.* [2]. They consider a scenario where nature randomly selects a game from a family of two-player matrix games, and informs player 1, but not player 2, about the selected game. The same game will now be played again and again. At each time unit $t = 1, 2, \dots$, the players choose their moves (actions) which collectively determine their payoffs, and both players observe each others actions. Player 1 is confronted with the dilemma of whether to play optimally in the game chosen by nature; if he does that (and if player 2 knows which policy is used by player 1), then player 2 will eventually be able to guess which is the game being played, so that player 1 loses his advantage of being informed. If, on the contrary, he uses a policy that does not utilize his knowledge of the game, then again he does not gain from being informed. Unlike in the game we consider, in this game the informed player does not directly control what information to reveal or to conceal to the other player. Also, here the information chosen by nature does not change with time, whereas we assume that the nature's choice changes with time and the evolution is temporally independent. Thus, here, unlike in the game we consider, at any given time a player can exploit the knowledge he has acquired from past interactions; in our case the game effectively starts fresh at each instant (our solutions therefore do not consider any temporal relation at all). Thus, the formal questions that are answered and also the techniques used to obtain the answers substantially differ in the two cases.

Information concealing has been extensively investigated in context of multi-media [10]. An example is the research on watermarking, where one tries to hide a signature in some picture or audio recording in order to be able to identify it later. Informally speaking, these scenarios consist of only one player who seeks to conceal as much information as possible. We consider a scenario with two players such that both players act sequentially and the first conceals information with the goal of degrading the performance of the second by decreasing the second's capabilities to make good action choices.

Again, the formal questions that are answered and also the techniques used to obtain the answers substantially differ in the two cases.

Finally, standard results in classical and computational game theory do not apply in the information concealing game we consider. First, classical game theory provides us with the PBE solution concept for signaling games [5], but does not guarantee uniqueness of this equilibrium. In our case, for any given pair of policies of the players, their utilities are functions of their informations, which in turn depend on the system states, and the system can be in one of several possible states. Next, general purpose algorithms are not known for computing a PBE policy-pair, either exactly or approximately with a provable-approximation guarantee, except when the PBE is the same as the well-known Nash equilibrium [3]¹, which happens in our case only when the number of system states is 1. We show that all PBE policy-pairs are functionally equivalent, and a PBE policy-pair can be obtained (exactly and not approximately) by solving linear programs with finite (but super-exponential in n) number of variables and constraints. The above results follow from a one-to-one correspondence that we have established between the PBE in the game we consider and the saddle-point strategies in an equivalent two-person zero-sum game. To the best of our knowledge, such equivalence is not commonly encountered in game-theory. Now, this equivalence does not however guarantee polynomial-time (polynomial in n) computation of equilibrium policies since the number of deterministic policies in the equivalent game is super-exponential in n in our case, which results in super-exponential number of variables and constraints of the above linear programs. Note that computational game theory focuses on determining exact solutions (e.g., for saddle-points of two-person zero-sum games Chapter III.2.4, [6]) whenever such solutions are computationally tractable, or approximations otherwise (e.g., for Nash equilibrium of bi-matrix games [4], [7]), using computation times that are polynomial in the number of deterministic policies of the players. Thus, since the number of deterministic policies is super-exponential in n in our case, standard algorithms will have computation times that are again super-exponential in n . To the best of our knowledge, standard algorithms for fast computation of exact solutions or approximations when the number of policies of the players is itself intractable (e.g., super-exponential) are not available in the literature. Thus, one of our important contributions has been to develop computationally efficient (that is with computation time that is polynomial in n) (i) exact solutions in special cases, and (ii) approximations with provable approximation guarantees in general cases using specialized arguments that exploit the above equivalence and the special characteristics of the game under consideration.

III. A MATHEMATICAL FRAMEWORK

We formulate the information concealing problem as a signaling game and consider the Perfect Bayesian Equilibrium or the PBE solution concept (Section III-A). We next elucidate the terminologies and the solution concept using the motivating examples presented in the previous section (Section III-B). We finally demonstrate that the PBE for this game exhibits several counter-intuitive properties which indicate that the computation of such equilibrium may not be straightforward (Section III-C). In Section VI, we generalize the framework to relax several assumptions made in this Section.

¹Nash equilibrium policies can be computed (i) exactly using a computation time that is exponential in the number of deterministic policies of the players (Chapter 3.4 [9]) (ii) approximately with provable approximation guarantees using a computation time that is polynomial in the number of deterministic policies of the players [4], [7].

A. Terminologies and Solution Concepts

We start by modeling the information concealing game as a stochastic leader-follower game between two players: the controller and the actor. We describe the game in both the normal form as well as in the strategic form.

- **System state:** The state of the system is an n -dimensional vector \vec{X} whose entries take values in $\mathcal{K} = \{0, \dots, K-1\}$. Let $\mathcal{N} = \{1, \dots, n\}$. The state space is \mathcal{K}^n . The random variables corresponding to the components of the state vector may be dependent and can be described by a joint probability distribution β .
- **Information of the Controller:** The controller knows the system state \vec{X} , and thereby has full information.
- **Actions of the Controller:** The controller conceals the values of at most k components of the system state from the actor; it decides which components it would conceal based on its information. Thus, the controller's action is a subset of \mathcal{N} with cardinality k or lower. Let $\mathcal{A}_c(\vec{x})$ denote the set of all sub-vectors of \vec{x} with $n-k$ or more components, and $\mathcal{A}_c = \cup_{\vec{x} \in \mathcal{K}^n} \mathcal{A}_c(\vec{x})$. We show in Section VI, the formulations and most of the results in this paper hold when we allow the controller to conceal exact values of all components in the entire system-state, and reveal arbitrary functions of the system state to the actor instead (e.g., the average of the states of the components, ranges containing the states of some components, etc.).
- **Information of the actor:** The actor knows the states of those components of the system state which the controller does not conceal. Specifically, if c is the action taken by the controller and the system state is \vec{x} , then the actor's information \vec{y} consists of the sub-vector of \vec{x} with components in $\mathcal{N} \setminus c$. Therefore, from its information \vec{y} , the actor knows the controller's action, i.e., the subset of components $a(\vec{y})$ the controller conceals.
- **Actions of the actor:** The actor selects one of the components of the system state. Thus, its action is an integer $l \in \mathcal{N}$. Again, we show in Section VI that the formulations and most of the results in this paper hold when we generalize the actions of the actor, that is, when the actor selects a sub-vector of the system state (instead of one component only).
- **Payoff function:** If a component of the system state has value i , then the expected utility associated with that component is $r(i)$ such that $r(0) < r(1) < \dots < r(K-1)$. If the system state is \vec{x} , and the actor selects component l , then its payoff is $r(x_l)$.
- **Common Knowledge:** The parameters $n, k, K, r(i)$ for each $i \in \mathcal{K}$ and β are common knowledge. These parameters are determined based on goals and constraints of specific systems (e.g., k may be determined based on resource constraints of the jammer in the cognitive radio network and the price the burglar has offered in the security system) - investigation of how these parameters are determined is beyond the scope of the current paper.
- **Strategies:**
 - **Pure policies:** A controller's (actor's, respectively) pure policy is a function from \mathcal{K}^n to \mathcal{A}_c (\mathcal{A}_c to \mathcal{N} , respectively). Let \mathcal{U}^p (\mathcal{V}^p , respectively) be the set of pure policies for the controller (actor, respectively).
 - **Mixed policies:** A mixed policy of a player is a probability measure over its pure policies. Let \mathcal{U} (\mathcal{V} , respectively) be the set of mixed policies for controller (actor, respectively). Note that each pure policy of a player can be viewed as a (degenerate) mixed policy for the same player. A

policy u in \mathcal{U} (v in \mathcal{V}) can also be represented as the probability distribution $\{u(\vec{x})\}$ ($v(\vec{y})$, respectively) used by the controller (actor, respectively) for selecting its actions when its information is \vec{x} (\vec{y} , respectively). Specifically, $u(\vec{x})_{\vec{y}}$ ($v(\vec{y})_i$, respectively) is the probability with which the controller (actor, respectively) reveals the sub-vector $\vec{y} \in \mathcal{A}_c(\vec{x})$ (selects the component $i \in \mathcal{N}$, respectively) when its information is \vec{x} (\vec{y} , respectively).

Let $E_{\beta}^{u,v}$ be the expectation operator for the action and informations of the two players when the players use policies $u \in \mathcal{U}$, $v \in \mathcal{V}$ and β is the probability distribution of the system state.

- **Utility:** The utility of each player is its expected payoff conditioned on its information, and is therefore a function of its information.

- **Utility of the actor:** When the actor's information is \vec{y} , the controller and the actor use (behavioral or mixed) policies u and v respectively, and the joint probability distribution of the system state is β , the actor's utility $J_a^{\beta,u,v}(\vec{y})$ is given by

$$J_a^{\beta,u,v}(\vec{y}) = E_{\beta}^{u,v}[r(X_B)|\vec{Y}_a = \vec{y}], \quad (1)$$

where \vec{Y}_a is the random information of the actor, X_i is the random state of the i th component of the system state, B is the (potentially random) action of the actor.

- **Utility of the controller:** The controller's utility is the negative of the expected payoff of the actor conditioned on the controller's information. Specifically, when the system state is \vec{x} , and the controller and the actor use (behavioral or mixed) policies u and v respectively, the controller's utility $J_c^{u,v}(\vec{x})$ is given by

$$J_c^{u,v}(\vec{x}) = -E^{u,v}[r(x_B)|\vec{X} = \vec{x}], \quad (2)$$

where \vec{X} is the random system state, x_B is the B th component of \vec{x} , B is the (potentially random) action of the actor. This expectation depends on β only through u, v .

Thus, for any given policy-pair of the players, the utility of each player is a function of its information, rather than a single number. Also, note that the utility functions are quite general, except for the special relation we assume between the utilities of the two players, that is, that the controller's utility is the negative of the expected payoff of the actor conditioned on the controller's information; the payoff function of the actor can be arbitrary. This relation between the utility functions of the players has been motivated by our requirement that the players' utilities oppose each other and if one player's utility is high, the other's utility must be low. This relation will be key in computing the solutions of this game.

- **Controller's and Actor's goals:** The controller and the actor seek to maximize their respective utilities $J_c^{u,v}(\vec{x})$, $J_a^{\beta,u,v}(\vec{y})$ for all values of their respective informations \vec{x}, \vec{y} .

We now define the Perfect Bayesian Equilibrium (PBE) solution concept ([5], Chapter 8.2).

Definition 3.1: Let u^* and v^* be mixed policies of the controller and actor respectively. Then (u^*, v^*) is a Perfect Bayesian Equilibrium if the following conditions hold:

- for each $\vec{x} \in \mathcal{K}^n$ such that $\beta(\vec{x}) > 0$, $u^*(\vec{x})$ is a best response of the controller against v^* of the actor, i.e., $u^*(\vec{x})$ maximizes $J_c^{u,v^*}(\vec{x})$ among all $u \in \mathcal{U}$, and
- for each $\vec{y} \in \mathcal{A}_c$ which occurs with positive probability under $\beta, u^*, v^*(\vec{y})$ is a best response of the actor against u^* of the controller, i.e., $v^*(\vec{y})$ maximizes $J_a^{\beta,u^*,v}(\vec{y})$ among all $v \in \mathcal{V}$.

B. Elucidating examples

We now elucidate the above terminologies using the examples in Section I-B.

In cognitive radio networks the system state constitutes the states of the channels, and $r(i)$ is the expected rate of successful transmission of the transmitter (actor) when it transmits in a channel that is in state i . The jammer's (controller's) action is to conceal the states some ($\leq k$) channels and the transmitter's action is to select a channel for transmission. An example class of policies of the jammer, denoted as *Greedy for Controller* or GC, is to conceal the channels with k -best states, that is, those with k -best expected rates of successful transmission. u^{GC} denotes an arbitrary policy in this class. An example class of policies of the transmitter, denoted as *Best Among Revealed for Actor* or BRA, is to select the channel that has the highest state among the revealed channels. The pure policies in these classes are those that break the ties in some deterministic order. Let the jammer and transmitter use policies u, v respectively. Then, (a) $J_c^{u,v}(\vec{x})$ is the negative of the expected rate of successful transmission of the transmitter when the channel state is \vec{x} , and (b) $J_a^{\beta,u,v}(\vec{y})$ is the expected rate of successful transmission of the transmitter when the jammer reveals \vec{y} to the transmitter, and the joint distribution of the channel state is β . Consider the *Uniform among Concealed for Actor* or UCA policy of the transmitter that selects a channel for transmission uniformly among those whose states are concealed. Then $J_c^{u,\text{UCA}}(\vec{x}) = -\frac{1}{k} \max_{S \subseteq N, |S|=k} \sum_{i \in S} x_i$, and $J_a^{\beta,u,\text{UCA}}(\vec{y}) = \frac{1}{k} \sum_{i \in a(\vec{y})} E_{\beta}^{u,\text{UCA}}(X_i | \vec{y})$. If the transmitter uses a policy in the class BRA, any policy in the class GC is the jammer's best response, and if the state processes of the channels are identically distributed, UCA is the transmitter's best response against the GC policy of the jammer that breaks ties uniformly and randomly among the channels.

In the authentication example for e-commerce, the seller (actor) may have different bargaining powers associated with different informations it can learn about the buyer (controller), and the buyer may not know the seller's bargaining power associated with any piece even though he knows the details about the piece. This is because different sellers may have access to different data bases and therefore may be able to extract different amount of additional information about the buyer from the same content. The buyer may however know the expected bargaining power of the seller associated with each piece of information. This scenario can be modelled by assuming that each different piece of information of the buyer can be in one of K states and the knowledge of the state of a piece of information implies the knowledge of the expected and not the exact value of the bargaining power associated with that piece. Now, $r(i)$ is the expected bargaining power associated with a piece when it is in state i . The system state consists the states of the n pieces of informations the buyer has about himself. The action of the buyer is to reveal limited information about some ($n - k$) pieces of information in the first stage of the authentication: the seller can only determine \vec{y} the states of these pieces of information from the limited information the buyer reveals (since although he knows what databases he can search he does not know the details about any of these pieces). Let the buyer and the seller use policies u, v respectively. The seller's action is to select one piece for which it requests details. Then, (a) $J_c^{u,v}(\vec{x})$ is the negative of the expected bargaining power of the seller when the system state is \vec{x} , and (b) $J_a^{\beta,u,v}(\vec{y})$ is the expected bargaining power of the seller when it observes \vec{y} in the first stage, and the joint distribution of the system state is β .

In the gambling game, β can be obtained from the distribution that is simultaneously used to draw the random numbers, and K is the cardinality of the support set of this original distribution. Note that the

random numbers drawn may be negative; we enumerate them using K positive integers, and each such enumeration constitutes the state of a card. Thus, each card has K possible states, and $r(i)$ is the number associated with the i th state. The system state consists the random numbers on the cards drawn by the second gambler (actor), and is known only to the first. The action of the first gambler (controller) is to reveal the states of some ($\geq n - k$) of these cards, which constitutes the information \vec{y} for the second. The second gambler's action is to select one card among those that it selected initially, and his payoff is the number on this card. Let the gamblers use policies u, v respectively. Then, (a) $J_c^{u,v}(\vec{x})$ is the negative of the expectation of the random number on the card the second finally (potentially randomly) selects for examination when the system state is \vec{x} , and (b) $J_a^{\beta,u,v}(\vec{y})$ is the expectation of the number on the card the second finally selects for examination, when it observes \vec{y} and the joint distribution of the system state is β .

The query resolution network and the security systems are similar to the cognitive radio network. In the former, the system state constitutes the states of the databases, each database can be in K states, and $r(i)$ is the probability that the information sought is in a database that is in state i . In the latter, the system state constitutes the states of the gates (e.g., the number of guards at each gate), each gate can be in K states, each state represents a level of efficacy of the security system at the gate and $r(i)$ is the probability that the burglar will successfully break in through a gate that is in state i .

C. Counter-intuitive properties of the Perfect Bayesian Equilibrium

We now demonstrate that the PBE exhibits several counter-intuitive properties which suggests that it may not always consist of simple policies that can be represented in closed form. This in turn motivates the design of efficient frameworks for computing it, which is the focus of the next two sections.

Consider the "Greedy for Controller" (GC) class of policies for the controller (Section III-B). The policies in this class conceal the components with k highest states. Intuitively, it seems that some GC policy minimizes the efficacy of the actor and therefore there always exists some GC policy and some policy v for the actor such that the pair is a point-wise nash equilibrium. The following lemma shows that this intuition is unfounded, even when the state processes for different components are mutually independent and identically distributed (i.e., even when all channels are i.i.d. in cognitive radio networks).

Lemma 3.1: There may not exist any policy u in the class GC, and $v \in \mathcal{V}$ such that (u, v) is a PBE, even in systems where the state processes for different components are mutually independent and identically distributed.

Next, consider a simple class of policies "Statistically Best for Actor" (SBA) for the actor under which when its information is \vec{y} , it selects a component i for which $E_{\beta}[r(X_i) | \vec{Y} = \vec{y}]$ is the maximum. Again, different policies in this class use different tie-break rules. Note that the above conditional expectation is computed using only β and not the controller's policy. For example, when the state processes of all components are mutually independent, $K = 2$, component i is in state j with probability p_{ij} , $r(0) = 0$, a policy in SBA selects a component that is in state 1 if the state of a component that is in state 1 has been revealed, and selects a concealed component i for which p_{i1} is the maximum. We will use v^{SBA} to denote an arbitrary policy in this class. It may seem that at least in simple special cases, i.e., when $K = 2$, there always exists some v^{SBA} such that (u, v^{SBA}) is a PBE for some policy u of the controller. The following lemma shows that such intuition is unfounded.

Lemma 3.2: There may not exist a policy pair $u \in \mathcal{U}, v^{\text{SBA}} \in \text{SBA}$ such that (u, v^{SBA}) is a PBE, even in systems where the state

processes for different components are mutually independent and $K = 2$.

We prove lemmas 3.1 and 3.2, in appendices A and B respectively, after obtaining some additional properties of the PBE.

IV. A COMPUTATIONAL FRAMEWORK FOR THE PERFECT BAYESIAN EQUILIBRIUM

The signaling game formulated in the previous section is clearly not a two-person zero-sum game as the arguments of the controller's and actor's utility functions have different dimensions, and hence the sum of these functions is not well-defined. Nevertheless, owing to the relations between the players' utilities ((1) and (2)) we demonstrate that there exists an equivalent zero-sum game with finitely many pure policies for each player such that a policy pair (u, v) of the controller and actor is a PBE in the original game if and only if it is a saddle point in the equivalent game (Section IV-A). This equivalence implies that all PBE policy-pairs are functionally equivalent in the original game, and one such equilibrium can be determined by solving a pair of linear programs. The number of variables and constraints of this linear program is however super-exponential in n , and hence the linear program turns out to be computationally intractable even for moderate n . Nevertheless, using this equivalence, we next develop a framework for computing the PBE using a computation time which is exponential in n (Section IV-B).

A. An equivalent two-person zero-sum game

Definition 4.1: Consider a game with two players: the controller and the actor. The action of each player now is to select one of its pure policies in the signaling game described in the previous section. When the two players select policies u, v respectively, the utility of the actor under the joint probability distribution β for the system states is given by

$$R_\beta^{u,v} = E_\beta^{u,v}[r(X_B)] = \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) E_\beta^{u,v}[r(x_B) | \vec{X} = \vec{x}]. \quad (3)$$

where B is the action of the actor under policies u, v and random system state \vec{X} . The actor seeks to maximize its utility and the controller seeks to minimize the actor's utility. The game is clearly a two-person zero-sum game with finitely many pure policies for each player.

For notational simplicity, we use the same notations (e.g., $u, v, \mathcal{U}, \mathcal{V}$, etc.) to denote the individual mixed policies and the sets of mixed policies in both games. Clearly,

$$R_\beta^{u,v} = - \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) J_c^{u,v}(\vec{x}) \quad \forall u, v, \beta, \quad (4)$$

$$\text{and } R_\beta^{u,v} = \sum_{\vec{y} \in \mathcal{A}_c} \text{Pr}^{\beta,u}(\vec{y}) J_a^{\beta,u,v}(\vec{y}) \quad \forall u, v, \beta. \quad (5)$$

Thus, although the utilities of the controller and actor in the original game are functions of their informations, utility of the actor in the above two-person zero sum game is a number, which turns out to be (a) the negative of the expectation of the utility of the controller in the original game over all system states (which are the controller's information) (from equation (4)), and also (b) the expectation of the utility of the actor in the original game over all possible information vectors of the actor (from equation (5)).

Definition 4.2: The upper and lower values, $\bar{R}_\beta, \underline{R}_\beta$ of the above two-person zero-sum game are

$$\bar{R}_\beta = \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u,v}, \quad \underline{R}_\beta = \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u,v}.$$

Thus, \bar{R}_β , referred to as the min-max utility of the actor, is the minimum possible utility of the actor in the two-person zero-sum

game if it selects its policy so as to maximize its utility while assuming full knowledge of the controller's policy. Also, \underline{R}_β , referred to as the max-min utility of the actor, is the maximum possible utility of the actor in the two-person zero-sum game if the controller selects its policy so as to minimize the actor's utility while assuming full knowledge of the actor's policy.

For any $u^* \in \mathcal{U}$ and $v^* \in \mathcal{V}$ we have

$$\inf_{u \in \mathcal{U}} R_\beta^{u,v^*} \leq \underline{R}_\beta \leq \bar{R}_\beta \leq \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}. \quad (6)$$

Definition 4.3: If for some $u^* \in \mathcal{U}$ and $v^* \in \mathcal{V}$, $\inf_{u \in \mathcal{U}} R_\beta^{u,v^*} = \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}$ then all inequalities in (6) hold with equality and u^* (v^* , respectively) is called the saddle point policy of the controller (actor, respectively).

If u^*, v^* are saddle point policies of the controller and actor respectively, $\inf_{u \in \mathcal{U}} R_\beta^{u,v^*} = R_\beta^{u^*,v^*} = \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}$, and hence $R_\beta^{u^*,v^*} = \bar{R}_\beta = \underline{R}_\beta$. Thus, $R_\beta^{u^*,v^*}$ is denoted as the value of the two-person zero-sum game. Also, if both the controller and the actor selects the saddle-point policies, the actor's utility in the two-person zero-sum game equals its max-min and min-max utilities.

Two-person zero-sum games, with finitely many pure policies for each player, are known to have a saddle point within the mixed policies (Chapter III.2.4, [6]). The following theorem proves that a pair of policies constitutes a saddle point for the two-person zero-sum game if and only if it is a PBE of the original game.

Theorem 4.1: A mixed policy pair (u^*, v^*) is a PBE in the original game if and only if the corresponding mixed policy pair (u^*, v^*) is a saddle point pair in the two-person zero-sum game.

This theorem holds because of the relation between the utilities of the controller and actor we consider, that is, since the controller's utility is the negative of the expected payoff of the actor conditioned on the controller's information ((1) and (2)). Such equivalence is not true for arbitrary signaling games, or even for arbitrary "partial zero-sum games" [1] of which our game is a special case. Partial zero sum games are those that have a basic zero-sum feature: the sum of utilities for the two players that correspond to a fixed action-pair and system state, is zero, but are not zero-sum games since the players have different information. Aumann *et al.* [1] showed that such games may lead to equivalent games that are not zero-sum. Hence, although the transformation that we use is quite standard, the fact that it leads to a zero-sum game is new and specific to our problem.

Proof: Assume that (u^*, v^*) is a PBE. We show that it is a saddle point pair. From definition 4.3 and since there always exists a saddle point pair in the two-person zero-sum game, the above is indeed the case if (i) u^* minimizes R_β^{u,v^*} and (ii) v^* maximizes $R_\beta^{u^*,v}$. We show that (i) holds. Assume it does not. Then for some u , $R_\beta^{u,v^*} < R_\beta^{u^*,v^*}$. Hence, from (4), there exists some $\vec{x} \in \mathcal{K}^n$ such that $J_c^{u,v^*}(\vec{x}) > J_c^{u^*,v^*}(\vec{x})$ and $\beta(\vec{x}) > 0$. This contradicts the assumption that (u^*, v^*) is a PBE. Thus, (i) holds. Using (5), it can be similarly shown that (ii) holds as well. Thus, (u^*, v^*) is a saddle point pair.

Conversely, assume that (u^*, v^*) is a saddle point pair. We show that (i) in Definition 3.1 holds. Assume it does not. Then for some \vec{x} and u , $J_c^{u,v^*}(\vec{x}) > J_c^{u^*,v^*}(\vec{x})$ and $\beta(\vec{x}) > 0$. Define the policy w for the controller as the one that coincides with u if the system state is \vec{x} and that coincides otherwise with u^* . Then $R_\beta^{w,v^*} < R_\beta^{u^*,v^*}$. This contradicts the assumption that (u^*, v^*) is a saddle point pair. Thus, (i) holds. It can be similarly shown that (ii) holds as well. Thus, (u^*, v^*) is a PBE. ■

Theorem 4.1 constitutes the basis for proving the counter-intuitive properties of the PBE described in Section III-C. For example, for

proving lemma 3.1, we show that when $K \geq 3$, no GC policy may constitute a saddle point for the controller. This is because if the actor knows that the controller is using a GC policy, it also knows that any component whose state has been concealed is in a state which is at least as good as that of a component whose state has been revealed, and thus, its best action is to select a channel whose state has been concealed. Now, if instead of using a GC policy, the controller reveals the states of some components whose states are better than those of the components whose states he conceals, the actor may be confused regarding the choice of the component even when it knows the controller's policy, and is therefore more likely to make a poor selection. For example, when $K = 3$, if the controller reveals some components in state 1 and conceals some components in state 0, the actor may select a concealed component hoping that it is in state 2, and the component may instead be in state 0. This is however not the case when $K = 2$ (Observation 2). This is because now the components are in states 1 or 0. Thus, the controller can not confuse the actor by revealing some component that are in state 1, as then the actor will select the revealed component since it knows that no other component can be in a better state.

We next argue that a PBE exists in this original game, and that although the PBE policy-pairs are not-necessarily unique, all PBE policy-pairs are however functionally equivalent in the following sense:

Corollary 4.1: A PBE (u^*, v^*) exists in the original game. All PBE policy-pairs in the original game fetch in the original game (i) the same expected utility over all system states for the controller and (ii) the same expected utility over all possible information vectors for the actor.

Proof: The first statement follows since the information revealing game is a signaling game with a finite number of players, policies and system states. Such signaling games, referred to as finite signaling games, always have at least one PBE [8]².

The second statement follows from Theorem 4.1 since (i) any PBE policy-pair constitutes a saddle-point in the equivalent two-person zero-sum game, (ii) any saddle-point policy pair fetches the same utility, \bar{R}_β , for the actor in the equivalent game and (iii) the utility of the actor in the equivalent game under any policy-pair equals the expectation of the utility of the actor, and the negative of the expectation of the utility of the controller, in the original game under the same policy pair (equations (4) and (5) and the discussion immediately after).

Henceforth, we focus on the properties and computations of the saddle point. Also, owing to the equivalence of the saddle-point policies in the two games and since the utilities of the players in the original games are vectors, while the utility of the actor in the two-person zero-sum game is a number which has a simple linear relation with (that is, either positive or negative of, depending on the player as discussed after equations (4) and (5)) the expectations of the utilities of both players in the original game, we will henceforth focus on the utility of the actor in the two-person zero-sum game. Specifically, whenever we refer to the utility of the actor, we will refer to that in the two-person zero-sum game, unless stated otherwise.

B. Computation of the saddle point policies

We now investigate the computation of saddle point policies. It is well-known that a saddle point policy of a player in a two-person

²In our context, this statement also independently follows from Theorem 4.1 and since two-person zero-sum games, with finitely many pure policies for each player, are known to have a saddle point within the mixed policies (Chapter III.2.4, [6]).

zero-sum game can be computed using a linear program whose number of variables equal the number of its pure policies and the number of constraints equal the number of pure policies of the other player (Chapter III.2.4, [6]). This may sound quite encouraging at first since solving linear programs is polynomial in the number of decision variables and constraints. Nevertheless, the computation is intractable due to the huge number of pure policies N_c of the controller and N_a of the actor, given by

$$N_c = \left(\sum_{i=0}^k \binom{n}{i} \right)^{K^n} \quad \text{and} \quad N_a = n \sum_{i=0}^k \binom{n}{i} K^{n-i}. \quad (7)$$

(7) is obtained as follows.

- The controller's information has K^n possible values, and for each such information it can choose $\sum_{i=0}^k \binom{n}{i}$ actions (note that $\sum_{i=0}^k \binom{n}{i}$ is the number of subsets of the components of cardinality at most k).
- The actor's information has $\sum_{i=0}^k \binom{n}{i} K^{n-i}$ possible values, and for each such information it can choose n actions.

Simplifying (7), the number of pure policies of the controller (actor, respectively) in the original game is at least $\binom{n}{k}^{K^n}$ ($n^{\min(\lfloor n/2 \rfloor, K^{\lfloor n/2 \rfloor})}$, respectively). The computation is therefore intractable even for moderate values of n, K .

Exploiting system characteristics, we however compute the saddle point policies using linear programs whose number of variables and constraints are substantially fewer than those of the linear programs $((K^n k) \binom{n}{k})$ as opposed to N_c and N_a above) which the generic theory for two-person zero sum games provide. Specifically, the computation times of the linear programs we develop are polynomials in $(K^n k) \binom{n}{k}$, and therefore substantially lower than that of the generic linear programs.

Henceforth, u (v , respectively) denotes the probabilities with which the controller (actor, respectively) select the actions given their informations. Specifically, $u(\vec{x})_{\vec{y}}$ ($v(\vec{y})_i$, respectively) is the probability with which the controller (actor, respectively) reveals the sub-vector $\vec{y} \in \mathcal{A}_c(\vec{x})$ (selects the component $i \in \mathcal{N}$, respectively) when its information is \vec{x} (\vec{y} , respectively). Each such probability distribution corresponds to a mixed policy for the respective player. Hence, with slight abuse of notations, we state that $u \in \mathcal{U}$ and $v \in \mathcal{V}$.

1) *Saddle point for the controller:* The following linear program obtains a saddle point policy for the controller.

$$\begin{aligned} \text{LP-CONTROLLER:} \quad & \text{Min}_{\{z(\vec{y}), u(\vec{x})_{\vec{y}}\}} \sum_{\vec{y} \in \mathcal{A}_c} z(\vec{y}) \quad \text{s.t.} \\ & z(\vec{y}) \geq \sum_{\vec{x}: \vec{y} \in \mathcal{A}_c(\vec{x})} \beta(\vec{x}) r(x_i) u(\vec{x})_{\vec{y}} \\ & \quad \quad \quad \forall i \in \mathcal{N}, \vec{y} \in \mathcal{A}_c \\ \sum_{\vec{y} \in \mathcal{A}_c(\vec{x})} u(\vec{x})_{\vec{y}} & = 1 \quad \text{for all } \vec{x} \in \mathcal{K}^n \\ u(\vec{x})_{\vec{y}} & \geq 0 \quad \forall \vec{x} \in \mathcal{K}^n, \vec{y} \in \mathcal{A}_c(\vec{x}) \end{aligned}$$

Theorem 4.2: Any optimum solution $\{u(\vec{x})_{\vec{y}}\}_{\vec{y} \in \mathcal{A}_c(\vec{x}), \vec{x} \in \mathcal{K}^n}$ of LP-CONTROLLER is a saddle point policy u^* for the controller.

We first provide the intuition behind the proof. Note that $z(\vec{y})$ is the product of (i) the probability that the controller reveals \vec{y} to the actor and (ii) the maximum possible utility of the actor if the controller uses policy u and reveals \vec{y} to the actor. The following theorem will prove that the saddle-point policy of the controller is the one that minimizes the sum of $z(\vec{y})$ over the set of all possible information vectors of the actor. The constraints of the above linear program can be motivated by the following observations. The right hand side of the first constraint is the product of (i) the probability that the controller reveals \vec{y} to the actor and the (ii) utility of the actor

if it selects component i and the controller uses policy u and reveals \bar{y} to the actor. From the characterization of $z(\bar{y})$ in the second sentence of this paragraph, $z(\bar{y})$ must be at least the above quantity for each component i . Note that $\{u(\bar{x}_{\bar{y}})\}$ satisfies the last two constraints of the above linear program if and only if it is a policy of the controller. The formal proof follows.

Proof: From (5), for any $u \in \mathcal{U}, v \in \mathcal{V}, \beta$,

$$R_\beta^{u,v} = \sum_{\bar{y} \in \mathcal{A}_c} \Pr^{\beta,u}(\bar{y}) E_\beta^{u,v}[r(X_B)|\bar{Y} = \bar{y}].$$

Given $u \in \mathcal{U}$, consider a policy $v_u \in \mathcal{V}$ such that for each $\bar{y} \in \mathcal{A}_c$, $v_u(\bar{y})_j = 1$ for some j such that $E_\beta^u[r(X_j)|\bar{Y} = \bar{y}] = \max_{i \in \mathcal{N}} E_\beta^u[r(X_i)|\bar{Y} = \bar{y}]$, and $v_u(\bar{y})_j = 0$, for other values of j (i.e., under v_u w.p. 1 B is a component i that attains the above maximum and hence v_u is the actor's best response to controller's policy u). Note that

$$\begin{aligned} \max_{v \in \mathcal{V}} E_\beta^{u,v}[r(X_B)|\bar{Y} = \bar{y}] &= \max_{i \in \mathcal{N}} E_\beta^u[r(X_i)|\bar{Y} = \bar{y}] \\ &= E_\beta^{u,v_u}[r(X_B)|\bar{Y} = \bar{y}], \quad \forall \bar{y} \in \mathcal{A}_c. \end{aligned}$$

Thus,

$$\sup_{v \in \mathcal{V}} R_\beta^{u,v} = \sum_{\bar{y} \in \mathcal{A}_c} \Pr^{\beta,u}(\bar{y}) \max_{i \in \mathcal{N}} E_\beta^u[r(X_i)|\bar{Y} = \bar{y}] = R_\beta^{u,v_u}. \quad (8)$$

Hence,

$$\bar{R}_\beta = \inf_{u \in \mathcal{U}} R_\beta^{u,v_u}. \quad (9)$$

Next,

$$\begin{aligned} &E_\beta^u[r(X_i)|\bar{Y} = \bar{y}] \\ &= \sum_{\bar{x} \in \mathcal{K}^n} E_\beta^u[r(X_i)|\bar{Y} = \bar{y}, \bar{X} = \bar{x}] \Pr^{\beta,u}(\bar{X} = \bar{x}|\bar{Y} = \bar{y}) \\ &= \sum_{\bar{x} \in \mathcal{K}^n} r(x_i) \Pr^{\beta,u}(\bar{Y} = \bar{y}|\bar{X} = \bar{x}) \frac{\Pr^{\beta,u}(\bar{X} = \bar{x})}{\Pr^{\beta,u}(\bar{Y} = \bar{y})} \\ &= \sum_{\bar{x} \in \mathcal{K}^n} r(x_i) u(\bar{x})_{\bar{y}} \beta(\bar{x}) / \Pr^{\beta,u}(\bar{Y} = \bar{y}). \end{aligned} \quad (10)$$

Thus,

$$E_\beta^u[r(X_i)|\bar{Y} = \bar{y}] \Pr^{\beta,u}(\bar{Y} = \bar{y}) = \sum_{\bar{x} \in \mathcal{K}^n} r(x_i) u(\bar{x})_{\bar{y}} \beta(\bar{x}).$$

Thus, from (8) and (9),

$$\begin{aligned} R_\beta^{u,v_u} &= \sum_{\bar{y} \in \mathcal{A}_c} \max_{i \in \mathcal{N}} \sum_{\bar{x} \in \mathcal{K}^n} r(x_i) u(\bar{x})_{\bar{y}} \beta(\bar{x}) \\ \text{and } \bar{R}_\beta &= \inf_{u \in \mathcal{U}} \sum_{\bar{y} \in \mathcal{A}_c} \max_{i \in \mathcal{N}} \sum_{\bar{x} \in \mathcal{K}^n} r(x_i) u(\bar{x})_{\bar{y}} \beta(\bar{x}). \end{aligned}$$

Now, consider a feasible solution (u, z) of LP-CONTROLLER, such that z is chosen so as to minimize the value of the objective function subject to choosing u . The value of the objective function is R_β^{u,v_u} for any such pair.

Thus, if u^O is the optimum solution of LP-CONTROLLER, $\bar{R}_\beta = R_\beta^{u^O, v_{u^O}}$. Thus, from (8), $\bar{R}_\beta = \sup_{v \in \mathcal{V}} R_\beta^{u^O, v}$. Now, since a saddle point policy pair always exists, it follows from Definition 4.3 that any $u' \in \mathcal{U}$ for which $\bar{R}_\beta = \sup_{v \in \mathcal{V}} R_\beta^{u', v}$ is a saddle point policy of the controller. Thus, u^O is a saddle point policy of the controller. ■

The following corollary proves an intuitive property of saddle point policies of the controller, and will help reduce the number of variables of LP-CONTROLLER.

Corollary 4.2: There exists a saddle point policy u^* of the controller which always conceals the states of k components.

Proof: Consider an optimal solution (u, z) of LP-CONTROLLER which conceals the states of fewer than k components with positive

probability for one or more system states, that is, there exists $\bar{x} \in \mathcal{K}^n, \bar{y} \in \mathcal{A}_c(\bar{x})$ such that $u(\bar{x})_{\bar{y}} > 0$ and $|a(\bar{y})| < k$ (recall that $a(\bar{y})$ is the set of components the controller conceals when the actor's information is \bar{y}). Since (u, z) is an optimal solution, $z(\bar{y}') = \max_{i \in \mathcal{N}} \sum_{\bar{x}': \bar{y}' \in \mathcal{A}_c(\bar{x}')} \beta(\bar{x}') r(x'_i) u(\bar{x}')_{\bar{y}'} \quad \forall \bar{y}' \in \mathcal{A}_c$. We will show that there exists another optimal solution of LP-CONTROLLER which always conceals the states of k components.

Consider a sub-vector of \bar{y}, \bar{w} , such that $|a(\bar{w})| = k$. Consider a new feasible solution (u', z') of LP-CONTROLLER such that for each $\bar{x}' \in \mathcal{K}^n, \bar{y}' \in \mathcal{A}_c(\bar{x}')$,

$$u'(\bar{x}')_{\bar{y}'} = \begin{cases} u(\bar{x}')_{\bar{y}} + u(\bar{x}')_{\bar{w}} & \text{if } \bar{y}' = \bar{w}, \bar{y} \in \mathcal{A}_c(\bar{x}') \\ 0 & \text{if } \bar{y}' = \bar{y}, \\ u(\bar{x}')_{\bar{y}'} & \text{otherwise.} \end{cases} \quad (11)$$

In words, u' is the same as u except that it reveals \bar{w} to the actor whenever u reveals \bar{y} to the actor. Let, $z'(\bar{y}') = \max_{i \in \mathcal{N}} \sum_{\bar{x}': \bar{y}' \in \mathcal{A}_c(\bar{x}')} \beta(\bar{x}') r(x'_i) u'(\bar{x}')_{\bar{y}'} \quad \forall \bar{y}' \in \mathcal{A}_c$.

Here, (u', z') is feasible since $\bar{w} \in \mathcal{A}_c(\bar{x}')$ for all \bar{x}' such that $\bar{y} \in \mathcal{A}_c(\bar{x}')$.

Also,

$$\begin{aligned} &\{\bar{y}' : u'(\bar{x}')_{\bar{y}'} > 0 \text{ for some } \bar{x}' \in \mathcal{K}^n, \text{ and } |a(\bar{y}')| < k\} \\ &\subset \{\bar{y}' : u(\bar{x}')_{\bar{y}'} > 0 \text{ for some } \bar{x}' \in \mathcal{K}^n, \text{ and } |a(\bar{y}')| < k\}. \end{aligned} \quad (12)$$

Thus, u' conceals the states of $k - 1$ or fewer components with positive probability for strictly fewer system states than u does.

Clearly, $z'(\bar{y}') = z(\bar{y}')$ for all $\bar{y}' \notin \{\bar{y}, \bar{w}\}$, $z'(\bar{y}) = 0$ and $z'(\bar{w}) \leq z(\bar{w}) + z(\bar{y})$. Thus, the value of the objective function under (u', z') is not higher than that under (u, z) . Thus, (u', z') is also an optimal solution of LP-CONTROLLER.

Thus, due to (12), repeating this process we obtain an optimal solution (u^*, z^*) of LP-CONTROLLER such that $\{u^*(\bar{x}')_{\bar{y}'} > 0 \text{ for some } \bar{x}' \in \mathcal{K}^n, \text{ and } |a(\bar{y}')| < k\} = \emptyset$, i.e., u^* always conceals the states of k components. The result follows. ■

Now, consider the following definition.

Definition 4.4: Let $\mathcal{A}_{c,k} = \{\bar{y} : |a(\bar{y})| = k, \bar{y} \in \mathcal{A}_c\}$ and $\mathcal{A}_{c,k}(\bar{x}) = \mathcal{A}_{c,k} \cap \mathcal{A}_c(\bar{x})$.

Due to Corollary 4.2, we only need to consider the variables $z(\bar{y})$ such that $|a(\bar{y})| = k$. Also, note that for any \bar{y} and \bar{x} such that $\bar{y} \in \mathcal{A}_c(\bar{x}), x_i = y_i$ for any $i \in \mathcal{N} \setminus a(\bar{y})$. Thus, LP-CONTROLLER can be described as follows.

LP-CONTROLLER:	$\begin{aligned} &\text{Min}_{\{z(\bar{y}), u(\bar{x})_{\bar{y}}\}} \sum_{\bar{y} \in \mathcal{A}_c} z(\bar{y}) \text{ s.t.} \\ &\forall \bar{y} \in \mathcal{A}_{c,k} \quad z(\bar{y}) \geq \max_{i \in \mathcal{N} \setminus a(\bar{y})} r(y_i) \sum_{\bar{x}: \bar{y} \in \mathcal{A}_c(\bar{x})} \beta(\bar{x}) u(\bar{x})_{\bar{y}}, \\ &z(\bar{y}) \geq \sum_{\bar{x}: \bar{y} \in \mathcal{A}_c(\bar{x})} \beta(\bar{x}) r(x_i) u(\bar{x})_{\bar{y}}, \\ &\forall i \in a(\bar{y}), \bar{y} \in \mathcal{A}_{c,k}, \\ &\sum_{\bar{y} \in \mathcal{A}_{c,k}(\bar{x})} u(\bar{x})_{\bar{y}} = 1 \quad \forall \bar{x} \in \mathcal{K}^n \\ &u(\bar{x})_{\bar{y}} \geq 0 \quad \forall \bar{x} \in \mathcal{K}^n, \bar{y} \in \mathcal{A}_{c,k}(\bar{x}) \end{aligned}$
----------------	---

Here, the right hand side of the first constraint is the product of (i) the probability that the controller reveals \bar{y} to the actor and (ii) the utility of the actor if it selects the revealed component that has the highest state and the controller uses policy u and reveals \bar{y} to the actor. The right hand side of the second constraint is the product of (i) the probability that the controller reveals \bar{y} to the actor and (ii) the utility of the actor if it selects concealed component i and the controller uses policy u and reveals \bar{y} to the actor.

Henceforth, we will use this description of LP-CONTROLLER. Note that LP-CONTROLLER has $O(K^n \binom{n}{k})$ variables and $O(k \binom{n}{k} K^n)$

constraints. Thus, the computation time of this linear program is polynomial in $K^n k \binom{n}{k}$.

2) *Saddle point for the actor*: The following linear program obtains a saddle point policy for the actor.

$$\begin{aligned} \text{LP-ACTOR:} \quad & \text{Max}_{\{z(\vec{x}), v(\vec{y})_i\}} \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) z(\vec{x}) \\ z(\vec{x}) \leq & \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) \quad \forall \vec{y} \in \mathcal{A}_c(\vec{x}), \vec{x} \in \mathcal{K}^n \\ v(\vec{y})_j \geq & 0 \quad \forall \vec{y}, j \in \mathcal{N} \\ \sum_{j \in \mathcal{N}} v(\vec{y})_j = & 1 \quad \forall \vec{y} \in \mathcal{A}_c \end{aligned}$$

Theorem 4.3: The optimum solution $\{v(\vec{y})_i\}_{i \in \mathcal{N}, \vec{y} \in \mathcal{A}_c}$ of LP-ACTOR is a saddle point policy v^* for the actor.

We first provide the intuition behind the proof. Note that $z(\vec{x})$ is the minimum possible utility of the actor if it uses policy v and the state of the system is \vec{x} . Since \vec{x} is a random variable, so is $z(\vec{x})$. The following theorem will prove that the saddle-point policy of the actor is the one that maximizes the expectation of $z(\vec{x})$ over all possible system states \vec{x} . The constraints of the above linear program can be motivated by the following observations. The right hand side of the first constraint is the actor's utility when the system state is \vec{x} and the actor's information is \vec{y} and the actor uses the policy v . From the characterization of $z(\vec{x})$ in the second sentence of this paragraph, $z(\vec{x})$ must be at most the above quantity for each possible information of the actor, \vec{y} . This is because the controller can reveal any such \vec{y} to the actor. Note that v satisfies the last two constraints of the above linear program if and only if it is a policy of the actor. The formal proof follows.

Proof: From (4), for any $u \in \mathcal{U}, v \in \mathcal{V}, \beta$,

$$\begin{aligned} R_\beta^{u,v} &= \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) E_\beta^{u,v} [r(x_B) | \vec{X} = \vec{x}] \\ &= \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) \sum_{\vec{y} \in \mathcal{A}_c(\vec{x})} u(\vec{x})_{\vec{y}} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i). \end{aligned}$$

Consider a policy $u_v \in \mathcal{U}$ such that for each $\vec{x} \in \mathcal{K}^n$, $u_v(\vec{x})_{\vec{y}} = 1$ for some $\vec{y} \in \mathcal{A}_c(\vec{x})$ such that $\sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) = \min_{\vec{t} \in \mathcal{A}_c(\vec{x})} \sum_{i \in \mathcal{N}} v(\vec{t})_i r(x_i)$ and $u_v(\vec{x})_{\vec{y}} = 0$, for all other $\vec{y} \in \mathcal{A}_c(\vec{x})$.

Since $u(\vec{x})$ is a probability distribution on $\mathcal{A}_c(\vec{x})$,

$$\begin{aligned} & \inf_{u \in \mathcal{U}} \sum_{\vec{y} \in \mathcal{A}_c(\vec{x})} u(\vec{x})_{\vec{y}} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) \\ &= \min_{\vec{y} \in \mathcal{A}_c(\vec{x})} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) \\ &= \sum_{\vec{y} \in \mathcal{A}_c(\vec{x})} u_v(\vec{x})_{\vec{y}} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i). \end{aligned}$$

$$\text{Thus, } \inf_{u \in \mathcal{U}} R_\beta^{u,v} = \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) \min_{\vec{y} \in \mathcal{A}_c(\vec{x})} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) = R_\beta^{u_v, v} \quad (13)$$

(i.e., u_v is the controller's best response to actor's v). Now, $\underline{R}_\beta = \sup_{v \in \mathcal{V}} R_\beta^{u_v, v}$. Thus, from (13),

$$\underline{R}_\beta = \sup_{v \in \mathcal{V}} \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) \min_{\vec{y} \in \mathcal{A}_c(\vec{x})} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i).$$

Now, consider a feasible solution (v, z) of LP-ACTOR, such that z is chosen so as to maximize the value of the objective function subject to choosing v . The value of the objective function is $R_\beta^{u_v, v}$ for any such pair. Thus, if v^O is the optimum solution of LP-ACTOR, $\underline{R}_\beta = R_\beta^{u_v^O, v^O}$. Thus, from (13), $\underline{R}_\beta = \inf_{u \in \mathcal{U}} R_\beta^{u, v^O}$. Now, since a saddle point policy pair always exists, it follows from Definition 4.3

that any $v' \in \mathcal{V}$ for which $\underline{R}_\beta = \inf_{u \in \mathcal{U}} R_\beta^{u, v'}$ is a saddle point policy of the actor. Thus, v^O is a saddle point policy of the actor. ■

Definition 4.5: A policy $v \in \mathcal{V}$ of an actor is said to be *sensible* if it never selects a component whose state has been revealed and which is in a state that is lower than the highest state among the states of all components whose states have been revealed (i.e., $v(\vec{y})_i = 0$ if $i \notin a(\vec{y})$ and $y_i \neq \max_{j \in \mathcal{N} \setminus a(\vec{y})} y_j$).

Observation 1: Note that $R_\beta^{u, v^1} = R_\beta^{u, v^2}$ for any $u \in \mathcal{U}, v^1, v^2 \in \mathcal{V}$ such that $v^1(\vec{y})_i = v^2(\vec{y})_i$ for any $i \in a(\vec{y})$ and $\sum_{i: i \notin a(\vec{y}), y_i = j} v^1(\vec{y})_i = \sum_{i: i \notin a(\vec{y}), y_i = j} v^2(\vec{y})_i$ for each $j \in \{0, \dots, K-1\}$.

The following corollary proves an intuitive property of saddle point policies of the actor, and will help reduce the number of variables of LP-ACTOR.

Corollary 4.3: There exists a sensible saddle point policy v^* of the actor.

Proof: For any $i \in \mathcal{N} \setminus a(\vec{y})$, $x_i = y_i$ if $\vec{y} \in \mathcal{A}_c(\vec{x})$. Thus, the first constraint in LP-ACTOR can be written as $z(\vec{x}) \leq \gamma(v, \vec{y}) + \sum_{i \in a(\vec{y})} v(\vec{y})_i r(x_i)$ for all $\vec{y} \in \mathcal{A}_c(\vec{x})$, where $\gamma(v, \vec{y}) = \sum_{i \in \mathcal{N} \setminus a(\vec{y})} v(\vec{y})_i r(y_i)$. Given a feasible solution v , consider another feasible solution v' such that for each $\vec{y} \in \mathcal{A}_c$,

$$v'(\vec{y})_i = \begin{cases} v(\vec{y})_i & \text{if } i \in a(\vec{y}), \\ \sum_{j \in \mathcal{N} \setminus a(\vec{y})} v(\vec{y})_j & \text{for some } i \text{ s.t. } i \in \mathcal{N} \setminus a(\vec{y}) \text{ and} \\ & y_i = \max_{j \in \mathcal{N} \setminus a(\vec{y})} y_j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that v' is a sensible policy, and the maximum value of the objective function for v (the maximization is w.r.t. z) is not higher than that for v' . This is because $\gamma(v', \vec{y}) \geq \gamma(v, \vec{y})$ for each $\vec{y} \in \mathcal{A}$ and $\sum_{i \in \mathcal{N} \setminus a(\vec{y})} v'(\vec{y})_i r(x_i) = \sum_{i \in \mathcal{N} \setminus a(\vec{y})} v(\vec{y})_i r(x_i)$ for each \vec{x}, \vec{y} . The result follows. ■

Due to Corollaries 4.2 and 4.3 and the above observation, we only consider sensible saddle point policies v for the actor and variables $v(\vec{y})$ such that $|a(\vec{y})| = k$ and need to determine the components $v(\vec{y})_j$ for $j \in a(\vec{y})$. For any sensible saddle point policy v of the actor,

$$\begin{aligned} \sum_{i \in \mathcal{N}} v(\vec{y})_i r(x_i) &= \left(1 - \sum_{i \in a(\vec{y})} v(\vec{y})_i \right) \max_{i \in \mathcal{N} \setminus a(\vec{y})} r(y_i) \\ &+ \sum_{i \in a(\vec{y})} v(\vec{y})_i r(x_i), \end{aligned}$$

where the first component in the r.h.s arises due to the actor's selection of revealed components with the highest state only under such a policy and the second arises due to the actor's selection of concealed components. Thus, the r.h.s of the first component of LP-ACTOR can be modified, and the overall linear program can be re-written as follows.

$$\begin{aligned} \text{LP-ACTOR:} \quad & \text{Max}_{\{z(\vec{x}), v(\vec{y})_i\}} \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) z(\vec{x}) \\ z(\vec{x}) \leq & \left(1 - \sum_{i \in a(\vec{y})} v(\vec{y})_i \right) \max_{i \in \mathcal{N} \setminus a(\vec{y})} r(y_i) \\ & + \sum_{i \in a(\vec{y})} v(\vec{y})_i r(x_i) \quad \forall \vec{y} \in \mathcal{A}_{c,k}(\vec{x}), \vec{x} \in \mathcal{K}^n \\ v(\vec{y})_j \geq & 0, \quad \forall j \in a(\vec{y}), \vec{y} \in \mathcal{A}_{c,k} \\ \sum_{j \in a(\vec{y})} v(\vec{y})_j \leq & 1, \quad \forall \vec{y} \in \mathcal{A}_{c,k} \end{aligned}$$

Henceforth, we consider the above description for LP-ACTOR. Thus, LP-ACTOR has $O(K^n k \binom{n}{k})$ variables and $O(K^n k \binom{n}{k})$ constraints. Thus, the computation time of LP-ACTOR is polynomial in $(K^n k) \binom{n}{k}$.

V. PERFORMANCE GUARANTEES USING POLYNOMIAL TIME COMPUTATION

We have proved that the saddle point policies can be obtained by solving linear programs whose number of variables is exponential in n and polynomial in K . Using fast algorithms for solving linear programs, the saddle point policies can now be computed for moderate values of n but the computation will still be intractable for large n . We therefore focus on obtaining provable performance guarantees using polynomial time computable policies. We first consider the important special case where the system consists of few classes of components such that all components in each class are statistically identical and the number of states K is small (note that each class may have a large number of components and therefore n can be large). We prove that the saddle point policies can be computed in polynomial time in such systems (Section V-A). Specifically, when the system consists of M classes of components, the saddle point policies can be obtained by solving linear programs with $O(n^{2KM})$ variables and $O(n^{2KM})$ constraints for arbitrary n, K, k, M . Thus, when all components are statistically identical ($M = 1$), the computation time is polynomial in n , but exponential in K (note that K is small in most systems). The result is interesting given that some intuitive policies do not constitute saddle point policies even when all components are statistically identical (Lemma 3.1). We next show that provable approximation guarantees can be obtained in arbitrary systems using some simple policies that can be computed in almost linear time (either $O(n)$ or $O(n \log n)$) time (Section V-B).

A. Polynomial time computation of saddle point policies in systems with constant number of classes of components and constant number of states

We first formally define the notion of classes of components and motivate the investigation of the special case where the system consists of a few classes and few states for the components. We subsequently present a key technical property (Theorem 5.1) for systems with arbitrary number of classes of components and states (Section V-A.1). Using this property and some additional terminologies (Section V-A.2), we show how saddle point policies for the controller and actor can be computed in polynomial time when K, M are constant (Sections V-A.3 and V-A.4).

Definition 5.1: Let $\vec{x}^{i,j} \in \mathcal{K}^n$ be obtained by interchanging the i th and the j th components of $\vec{x} \in \mathcal{K}^n$. Let $\vec{y}^{i,j} \in \mathcal{A}_c$ be obtained as follows: (a) if $i, j \notin a(\vec{y})$, then $a(\vec{y}^{i,j}) = a(\vec{y})$, $y_i^{i,j} = y_j$, $y_j^{i,j} = y_i$, $y_l^{i,j} = y_l$, $l \notin a(\vec{y}) \cup \{i, j\}$ (b) if $i \in a(\vec{y})$, $j \notin a(\vec{y})$, then $a(\vec{y}^{i,j}) = a(\vec{y}) \cup \{j\} \setminus \{i\}$, $y_i^{i,j} = y_j$, $y_l^{i,j} = y_l$, $l \notin a(\vec{y}^{i,j}) \cup \{i\}$, (c) if $i \notin a(\vec{y})$, $j \in a(\vec{y})$, then $a(\vec{y}^{i,j}) = a(\vec{y}) \cup \{i\} \setminus \{j\}$, $y_j^{i,j} = y_i$, $y_l^{i,j} = y_l$, $l \notin a(\vec{y}^{i,j}) \cup \{j\}$, (d) $\vec{y}^{i,j} = \vec{y}$, otherwise.

Definition 5.2: Components i, j are said to be in the same class if $\beta(\vec{x}) = \beta(\vec{x}^{i,j})$ for all $\vec{x} \in \mathcal{K}^n$. Note that the membership in the same class is an equivalence relation and hence the classes constitute a partition of \mathcal{N} . Let the system consist of M classes, where $1 \leq M \leq n$. The classes are numbered as $1, \dots, M$, and n_i components are in class i where $\sum_{i=1}^M n_i = n$. Let $a(\vec{y}, i)$ be the set of components in class i that have been concealed when the actor's information is \vec{y} . Note that $a(\vec{y}) = \cup_{i=1}^M a(\vec{y}, i)$.

Note that M can be determined from β and hence is also known to both players.

Several systems have large number of components but small or moderate number of classes of components and states. For example, cognitive radio networks may have large number of channels, but often, many of these channels are statistically identical, and hence the number of classes of channels is often substantially less than the number of channels. Also, the total number of states of these channels

is likely to be moderate as well. Next, consider the gambling example (Section I-B). The cards that have the same color constitute the same class as the distributions of the random numbers are statistically identical for all cards of the same color. Usually, the number of colors, or more generally number of types of cards (e.g., aces, jokers, etc.) is small although the number of cards can be large.

We first present a key property of systems with arbitrary number of classes of components.

1) Symmetry among components in the same class:

Definition 5.3: Let u, v be behavioral policies of the controller and actor respectively and $i, j \in \mathcal{N}$. The mirror image w.r.t (i, j) of the policy u (v , respectively), $u^{i,j} \in \mathcal{U}$ ($v^{i,j} \in \mathcal{V}$, respectively) is a policy obtained as follows: $u^{i,j}(\vec{x})_{\vec{y}} = u(\vec{x}^{i,j})_{\vec{y}^{i,j}}$ ($v^{i,j}(\vec{y})_i = v(\vec{y}^{i,j})_j$ and $v^{i,j}(\vec{y})_j = v(\vec{y}^{i,j})_i$, respectively).

Intuitively, $u^{i,j}$ ($v^{i,j}$, respectively) treat i as j and j as i .

Definition 5.4: A policy $u \in \mathcal{U}$ ($v \in \mathcal{V}$, respectively) is said to be symmetric w.r.t (i, j) if $u = u^{i,j}$ ($v = v^{i,j}$, respectively). A policy $u \in \mathcal{U}$ ($v \in \mathcal{V}$, respectively) is said to be symmetric if it is symmetric w.r.t. each pair of components that are in the same class. Let $\mathcal{U}^s \subset \mathcal{U}$ and $\mathcal{V}^s \subset \mathcal{V}$ be the classes of all symmetric policies of the controller and actor respectively.

The following theorem shows the existence of a symmetric saddle point policy for each player.

Theorem 5.1: There exists a symmetric policy $u \in \mathcal{U}^s$ ($v \in \mathcal{V}^s$, respectively) for the controller (actor, respectively) such that u (v , respectively) is a saddle point policy of the controller (actor, respectively).

Proof: We prove the theorem for the controller, and the proof for the actor is similar. Let $\mathcal{S}^u \subseteq \mathcal{N} \times \mathcal{N}$ be the set of tuples (a, b) such that a, b are in the same class and u is not symmetric w.r.t. a, b . From the definition of a symmetric policy, u is symmetric (i.e., $u \in \mathcal{U}^s$), iff $\mathcal{S}^u = \emptyset$. From Theorem 4.2, it is sufficient to prove that if there exists an optimal solution u of LP-CONTROLLER such that $\mathcal{S}^u \neq \emptyset$, there exists an optimal solution \hat{u} of LP-CONTROLLER such that $\mathcal{S}^{\hat{u}} \subset \mathcal{S}^u$. Note that such a \hat{u} is symmetric w.r.t. a strictly larger set of tuples of components in the same class. Thus, repeating the process, we can obtain an optimal solution which is symmetric w.r.t. all components in the same class, and is therefore, a symmetric optimal solution by definition.

Thus, we now consider u an optimal solution of LP-CONTROLLER such that $\mathcal{S}^u \neq \emptyset$, and set to obtain an optimal solution \hat{u} of LP-CONTROLLER such that $\mathcal{S}^{\hat{u}} \subset \mathcal{S}^u$. Then $u^{a,b}$ is an optimal solution of LP-CONTROLLER for any pair of components a, b that are in the same class. Now, consider an arbitrary pair of components $i, j \in \mathcal{S}^u$, and a policy $\hat{u} \in \mathcal{U}$ such that $\hat{u}(\vec{x})_{\vec{y}} = \frac{u(\vec{x})_{\vec{y}} + u^{i,j}(\vec{x})_{\vec{y}}}{2}$ for each $\vec{x} \in \mathcal{K}^n$ and $\vec{y} \in \mathcal{A}_c(\vec{x})$. In other words, \hat{u} is the same as u except that it treats component i (j respectively) as u treats component j (i respectively) 50% of the time. Now, since \hat{u} is a linear combination of two optimal solutions of LP-CONTROLLER, u and $u^{i,j}$, \hat{u} is an optimal solution of LP-CONTROLLER. Next, $\hat{u}^{i,j}(\vec{x})_{\vec{y}} = \frac{u^{i,j}(\vec{x})_{\vec{y}} + u(\vec{x})_{\vec{y}}}{2} = \hat{u}(\vec{x})_{\vec{y}}$ for each $\vec{x} \in \mathcal{K}^n$ and $\vec{y} \in \mathcal{A}_c(\vec{x})$. Thus, $\hat{u}^{i,j} = \hat{u}$, and hence \hat{u} is symmetric w.r.t. (i, j) . Thus, $(i, j) \notin \mathcal{S}^{\hat{u}}$. Also, note that $u^{i,j}$, and clearly u , are symmetric w.r.t. all tuples $(a, b) \notin \mathcal{S}^u$. Thus, \hat{u} is clearly symmetric w.r.t. all such tuples, and no such tuple belongs in $\mathcal{S}^{\hat{u}}$. Thus, $\mathcal{S}^{\hat{u}} \subseteq \mathcal{S}^u \setminus \{(i, j)\}$. The result follows. ■

Using Theorem 5.1, we show that the computation time for LP-CONTROLLER and LP-ACTOR can be substantially reduced when M and K are small.

2) Additional Terminologies:

Definition 5.5: Let $\mathbf{I}(\vec{x})$ be a matrix with M rows and K columns and entries in $0, \dots, n$ such that $l(\vec{x})_{i,j}$ is the number of components of \vec{x} that are in class i and state j . Let $\mathcal{L} = \{\mathbf{1} : \mathbf{I}(\vec{x}) = \mathbf{1}, \vec{x} \in \mathcal{K}^n\}$.

Let $\mathbf{m}(\vec{y})$ be a matrix with M rows and K columns with entries in $0, \dots, n - |a(\vec{y})|$ such that $m(\vec{y})_{i,j}$ is the number of components of \vec{y} that are in class i and state j . Let $\mathcal{M}_{\vec{x}} = \{\mathbf{m} : \mathbf{m}(\vec{y}) = \mathbf{m}, \vec{y} \in \mathcal{A}_{c,k}(\vec{x})\}$. Note that $\mathcal{M}_{\vec{x}^1} = \mathcal{M}_{\vec{x}^2}$ if $\mathbf{l}(\vec{x}^1) = \mathbf{l}(\vec{x}^2)$. Let $\mathcal{M}_1 = \cup_{\vec{x} \in \mathcal{K}^n, \mathbf{l}(\vec{x})=1} \mathcal{M}_{\vec{x}}$, and $\mathcal{M} = \cup_{\mathbf{l} \in \mathcal{L}} \mathcal{M}_1$.

With slight abuse of notation, we have used \mathbf{l}, \mathbf{m} to denote both the functions and the values of the functions as well - the implication of specific usages are clear from the context. We will use \vec{l}, \vec{m} instead of \vec{x}, \vec{y} so as to substantially reduce the number of variables and constraints of LP-CONTROLLER.

Note that (a) $|\{\vec{y} : \mathbf{m}(\vec{y}) = \mathbf{m}, \vec{y} \in \mathcal{A}_{c,k}(\vec{x})\}|$ depends on \vec{x} only through $\mathbf{l}(\vec{x})$, and (b) $|\{\vec{x} : \mathbf{l}(\vec{x}) = \mathbf{l}, \vec{y} \in \mathcal{A}_c(\vec{x})\}|$ depends on \vec{y} only through $\mathbf{m}(\vec{y})$. Thus, we can introduce the following definitions.

Definition 5.6: Let $\Theta_1(\mathbf{l}, \mathbf{m})$ denote for one (representative) \vec{x} such that $\mathbf{l}(\vec{x}) = \mathbf{l}$ the number of \vec{y} in $\mathcal{A}_{c,k}(\vec{x})$ such that $\mathbf{m}(\vec{y}) = \mathbf{m}$. Let $\Theta_2(\mathbf{l}, \mathbf{m})$ denote the number of system state vectors \vec{x} such that (a) $\mathbf{l}(\vec{x}) = \mathbf{l}$ and (b) $\vec{y} \in \mathcal{A}_c(\vec{x})$ for one (representative) \vec{y} such that $\mathbf{m}(\vec{y}) = \mathbf{m}$. Let $\Theta_3(\mathbf{m}) = |\{\vec{y} \in \mathcal{A}_{c,k} : \mathbf{m}(\vec{y}) = \mathbf{m}\}|$, and $\Theta_4(\mathbf{l}) = |\{\vec{x} \in \mathcal{K}^n : \mathbf{l}(\vec{x}) = \mathbf{l}\}|$.

Note that both $\Theta_2(\mathbf{l}, \mathbf{m})\Theta_3(\mathbf{m})$ and $\Theta_1(\mathbf{l}, \mathbf{m})\Theta_4(\mathbf{l})$ constitute the number of tuples (\vec{x}, \vec{y}) such that $\vec{x} \in \mathcal{K}^n, \vec{y} \in \mathcal{A}_{c,k}(\vec{x})$ and $\mathbf{l}(\vec{x}) = \mathbf{l}, \mathbf{m}(\vec{y}) = \mathbf{m}$. Thus,

$$\Theta_2(\mathbf{l}, \mathbf{m})\Theta_3(\mathbf{m}) = \Theta_1(\mathbf{l}, \mathbf{m})\Theta_4(\mathbf{l})$$

Definition 5.7: Let

$$R_1(\mathbf{m}) = \max_{j: \sum_{i=1}^M m_{i,j} > 0} r(j),$$

$$\text{and } R_2(\mathbf{l}, \mathbf{m}, i) = \sum_{j=0}^{K-1} r(j) \frac{l_{i,j} - m_{i,j}}{n_i - \sum_{j=0}^{K-1} m_{i,j}}.$$

Note that $R_1(\mathbf{m})$ is the expected reward the actor obtains when its information is \vec{y} such that $\mathbf{m}(\vec{y}) = \mathbf{m}$ and it selects a component whose state has been revealed and whose state is the highest among those of the components whose states have been revealed. Also, $R_2(\mathbf{l}, \mathbf{m}, i)$ is the expected reward the actor obtains when its information is \vec{y} such that $\mathbf{m}(\vec{y}) = \mathbf{m}$, the system state is \vec{x} such that $\mathbf{l}(\vec{x}) = \mathbf{l}$ and it selects a component of class i uniformly among $a(\vec{y}, i)$.

Definition 5.8: Let $\mathcal{C}(\mathbf{m}), 1 \leq |\mathcal{C}(\mathbf{m})| \leq \min(k, M)$, be the set of classes for which at least one component's state has been concealed when the actor's information \vec{y} is such that $\mathbf{m}(\vec{y}) = \mathbf{m}$. Let $\Phi(\mathbf{m}, i)$ be the number of components of class i that have been concealed when the actor's information \vec{y} is such that $\mathbf{m}(\vec{y}) = \mathbf{m}$. Note that $\Phi(\mathbf{m}, i) = \sum_{j=0}^{K-1} m_{i,j}$, and $|\mathcal{C}(\mathbf{m})| = \sum_{i=1}^M \min(\Phi(\mathbf{m}, i), 1)$.

Finally, since $\beta(\vec{x}) = \beta(\vec{x}^{i,j})$ for all i, j that are in the same class, $\beta(\vec{x}^1) = \beta(\vec{x}^2)$ if $\mathbf{l}(\vec{x}^1) = \mathbf{l}(\vec{x}^2)$.

Definition 5.9: Let $\beta'(\mathbf{l})$ denote $\beta(\vec{x})$ for some (representative) $\vec{x} \in \mathcal{K}^n$ such that $\mathbf{l}(\vec{x}^1) = \mathbf{l}$, and $\beta''(\mathbf{l}) = \Theta_4(\mathbf{l})\beta'(\mathbf{l})$.

Thus, $\beta''(\mathbf{l})$ is the probability that the system is in a state \vec{x} such that $\mathbf{l}(\vec{x}) = \mathbf{l}$.

3) *Polynomial time computation of saddle point policy of controller for constant K, M :* We now consider the simplification of LP-CONTROLLER.

Note that u is symmetric if and only if $u(\vec{x}^1)_{\vec{y}^1} = u(\vec{x}^2)_{\vec{y}^2}$ whenever the following conditions hold: (a) $\mathbf{l}(\vec{x}^1) = \mathbf{l}(\vec{x}^2)$, (b) $\mathbf{m}(\vec{y}^1) = \mathbf{m}(\vec{y}^2)$ (c) $\vec{y}^1 \in \mathcal{A}_c(\vec{x}^1), \vec{y}^2 \in \mathcal{A}_c(\vec{x}^2)$. Let $u'(\mathbf{l})_{\mathbf{m}}$ denote $u(\vec{x})_{\vec{y}}$ for some (representative) $\vec{x} \in \mathcal{K}^n, \vec{y} \in \mathcal{A}_{c,k}(\vec{x})$ such that $\mathbf{l}(\vec{x}) = \mathbf{l}, \mathbf{m}(\vec{y}) = \mathbf{m}$. Thus, each $u \in \mathcal{U}^s$ is uniquely described by $u^s(\mathbf{l})_{\mathbf{m}}$ where $u^s(\mathbf{l})_{\mathbf{m}} = \Theta_1(\mathbf{l}, \mathbf{m})u'(\mathbf{l})_{\mathbf{m}}$. Also, $\{u^s(\mathbf{l})_{\mathbf{m}}\}_{\mathbf{m} \in \mathcal{M}_1, \mathbf{l} \in \mathcal{L}}$ is a symmetric policy for the controller if and only if $\sum_{\mathbf{m} \in \mathcal{M}_1} u'(\mathbf{l})_{\mathbf{m}} = 1$ for all $\mathbf{l} \in \mathcal{L}$ and $u'(\mathbf{l})_{\mathbf{m}} \geq 0 \forall \mathbf{m} \in \mathcal{M}_1, \mathbf{l} \in \mathcal{L}$.

We now state LP-CONTROLLER-CLASS that computes $\{u^s(\mathbf{l})_{\mathbf{m}}\}$ for a symmetric saddle point policy of the controller.

$$\begin{aligned} \text{LP-CONTROLLER-CLASS:} \quad & \text{Min}_{\{\eta(\mathbf{m}), u'(\mathbf{l})_{\mathbf{m}}\}} \sum_{\mathbf{m} \in \mathcal{M}} \eta(\mathbf{m}) \quad s.t. \\ & \forall \mathbf{m} \in \mathcal{M}, \eta(\mathbf{m}) \geq R_1(\mathbf{m}) \sum_{\mathbf{l} \in \mathcal{L}} \beta''(\mathbf{l}) u^s(\mathbf{l})_{\mathbf{m}} \\ & \forall \mathbf{m} \in \mathcal{M}, i \in \mathcal{C}(\mathbf{m}) \eta(\mathbf{m}) \geq \sum_{\mathbf{l} \in \mathcal{L}} \beta''(\mathbf{l}) u^s(\mathbf{l})_{\mathbf{m}} R_2(\mathbf{l}, \mathbf{m}, i) \\ & \sum_{\mathbf{m} \in \mathcal{M}_1} u'(\mathbf{l})_{\mathbf{m}} = 1 \quad \text{for all } \mathbf{l} \in \mathcal{L} \\ & u'(\mathbf{l})_{\mathbf{m}} \geq 0 \quad \forall \mathbf{m} \in \mathcal{M}_1, \mathbf{l} \in \mathcal{L}. \end{aligned} \tag{14}$$

Theorem 5.2: The optimum solution $\{u^s(\mathbf{l})_{\mathbf{m}}\}_{\mathbf{m} \in \mathcal{M}_1, \mathbf{l} \in \mathcal{L}}$ of LP-CONTROLLER-CLASS is a symmetric saddle point policy for the controller.

We first provide the intuition behind the proof. Note that since we focus on computing a symmetric saddle point policy of the controller, and since the components in the same class are statistically identical, we can consider the controller's and actor's information as \vec{l}, \vec{m} instead of \vec{x}, \vec{y} respectively. Now, $\eta(\mathbf{m})$ is the product of the probability that the controller reveals \vec{m} to the actor and the maximum possible utility of the actor if the controller uses policy u^s and reveals \vec{m} to the actor. Thus, $\eta(\mathbf{m})$ plays the role of $z(\vec{y})$ in LP-CONTROLLER (refer to the paragraphs just after the statement of Theorem 4.2 and the formulation of LP-CONTROLLER at the end of Section IV-B.1). Now, LP-CONTROLLER-CLASS seeks to compute the saddle-point policy u^s by minimizing the sum of $\eta(\vec{m})$ over the set of all possible information vectors \vec{m} of the actor, just as LP-CONTROLLER seeks to compute the saddle-point policy u of the controller by minimizing the sum of $z(\vec{y})$ over the set of all possible information vectors \vec{y} of the actor. The constraints of LP-CONTROLLER-CLASS can be motivated by relating them to those of LP-CONTROLLER formulated just before Section IV-B.2. The right hand side of the first constraint of LP-CONTROLLER-CLASS is the product of (i) the probability that the controller reveals \vec{m} to the actor and (ii) the utility of the actor if it selects the revealed component with the highest state and the controller uses policy u^s and reveals \vec{m} to the actor. The right hand side of the second constraint of LP-CONTROLLER-CLASS is the product of (i) the probability that the controller reveals \vec{m} to the actor and (ii) the utility of the actor if it selects a concealed component i and the controller uses policy u^s and reveals \vec{m} to the actor. Both of these are analogous with the r.h.s. of the corresponding constraints of LP-CONTROLLER. Again, analogous to the last two constraints of LP-CONTROLLER, u^s satisfies the last two constraints of LP-CONTROLLER-CLASS if and only if it is a policy of the controller. The formal proof is relegated to appendix C.

Thus, LP-CONTROLLER-CLASS has $O(n^{2KM})$ variables and $O(n^{2KM})$ constraints. Thus, the computation time of LP-CONTROLLER-CLASS is polynomial in n and exponential in K, M , and hence polynomial in n if K, M are constants. The computation time can be reduced further for $K = 2$. We first observe the following.

Observation 2: For $K = 2$, there exists a saddle point policy for the controller in the GC class.

Recall that the ties can be broken by policies in the GC class in several different ways and thus all members in the GC class need not be saddle points; a saddle point policy can be computed if the appropriate tie-break policy is determined. Also, for any policy of the controller, there exists a best response policy of the actor that selects a component whose state is revealed and which is in state 1 whenever $m_{i,1} > 0$ for some i . Due to these observations, LP-CONTROLLER-CLASS needs to consider $\eta(\mathbf{m}), u^s(\mathbf{l})_{\mathbf{m}}$ only for \mathbf{l}, \mathbf{m}

such that $\sum_{i=1}^M l_{i,1} \leq k, m_{i,1} = 0 \forall i \in \{1, \dots, M\}$. Thus, LP-CONTROLLER-CLASS has $O(k^M)$ variables and $O(k^M)$ constraints in this case.

4) *Polynomial time computation of saddle point policy of actor for constant K, M* : We now consider the computation of a symmetric saddle point policy for the actor. Note that the actor's policy v is symmetric if and only if $v(\bar{y}^1)_i = v(\bar{y}^2)_j$ whenever the following conditions hold: (a) $\mathbf{m}(\bar{y}^1) = \mathbf{m}(\bar{y}^2)$ (b) i, j are in the same class, and (b) either (i) $i \in a(\bar{y}^1), j \in a(\bar{y}^2)$, or (ii) $i \notin a(\bar{y}^1), j \notin a(\bar{y}^2), y_i^1 = y_j^2$.

Consider a $\mathbf{m} \in \mathcal{M}$ and a class $i \in \mathcal{C}(\mathbf{m})$. Then, let $v'(\mathbf{m})_i$ be the probability with which a symmetric policy v selects one (representative) component, say j , that is in class i and has been concealed, when the actor's information state is a (representative) \bar{y} such that $\mathbf{m}(\bar{y}) = \mathbf{m}$ (i.e., $v'(\mathbf{m})_i = v(\bar{y})_j$). Let $v^s(\mathbf{m})_j = \Phi(\mathbf{m}, j)v'(\mathbf{m})_j, j \in \mathcal{C}(\mathbf{m})$, be the total probability with which a symmetric policy $v \in \mathcal{V}^s$ of the actor selects a component which is in class j and whose state has been concealed, when the actor's information state is a (representative) \bar{y} such that $\mathbf{m}(\bar{y}) = \mathbf{m}$. Thus, v selects a component whose state has been revealed with probability $1 - \sum_{j \in \mathcal{C}(\mathbf{m}(\bar{y}))} v^s(\mathbf{m}(\bar{y}))_j$. From Corollary 4.3 it is sufficient to consider only sensible policies. Note that $v^s(\mathbf{m})_j, j \in \mathcal{C}(\mathbf{m})$ uniquely specifies a symmetric sensible saddle point policy $v \in \mathcal{V}^s$ for the actor. Also, any $\{v^s(\mathbf{m})_j\}_{\mathbf{m} \in \mathcal{M}, j \in \mathcal{C}(\mathbf{m})}$ that satisfies $v^s(\mathbf{m})_i \geq 0 \forall i \in \mathcal{C}(\mathbf{m}), \mathbf{m} \in \mathcal{M}, \sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i \leq 1 \forall \mathbf{m} \in \mathcal{M}$ provides a symmetric, sensible policy for the actor.

We prove that the following linear program, LP-ACTOR-CLASS, computes a symmetric, sensible saddle point policy for the actor.

LP-ACTOR-CLASS:	$\text{Max}_{\{\eta(\mathbf{l}), v^s(\mathbf{m})_i\}} \sum_{\mathbf{l} \in \mathcal{L}} \beta''(\mathbf{l}) \eta(\mathbf{l}) \quad s.t.$
	$\eta(\mathbf{l}) \leq \left(1 - \sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i R_1(\mathbf{m}) \right) + \sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i R_2(\mathbf{l}, \mathbf{m}, i)$
	$\forall \mathbf{m} \in \mathcal{M}_1, \mathbf{l} \in \mathcal{L}$
	$v^s(\mathbf{m})_i \geq 0 \quad \forall i \in \mathcal{C}(\mathbf{m}), \mathbf{m} \in \mathcal{M}$
	$\sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i \leq 1 \quad \forall \mathbf{m} \in \mathcal{M}$

Theorem 5.3: The optimum solution $\{v^s(\mathbf{m})_j\}_{\mathbf{m} \in \mathcal{M}, j \in \mathcal{C}(\mathbf{m})}$ of LP-ACTOR-CLASS is a symmetric saddle point policy for the actor.

We first provide the intuition behind the proof. Note that since we focus on computing a symmetric saddle point policy of the actor, and since the components in the same class are statistically identical, we can consider the controller's and actor's information as \bar{l}, \bar{m} instead of \bar{x}, \bar{y} respectively. Now, $\eta(\mathbf{l})$ is the minimum possible utility of the actor if it uses policy v and the state of the system is \bar{l} . Thus, $\eta(\mathbf{l})$ plays the role of $z(\bar{x})$ in LP-ACTOR (refer to the paragraphs just after the statement of Theorem 4.3 and just before the formulation of LP-ACTOR at the end of Section IV-B.2). Now, LP-ACTOR-CLASS seeks to compute the saddle-point policy v^s by maximizing the sum of $\eta(\bar{l})$ over the set of all possible system states \bar{l} , just as LP-ACTOR seeks to compute the actor's saddle-point policy v by minimizing the sum of $z(\bar{x})$ over the set of all possible system states \bar{x} . The constraints of LP-ACTOR-CLASS can be motivated by relating them to those of LP-ACTOR formulated at the end of Section IV-B.2. The right hand side of the first constraint of LP-ACTOR-CLASS is the actor's utility when the system state is \bar{l} and the actor's information is \bar{m} and the actor uses the policy v . This is analogous with the r.h.s. of the first constraint of LP-ACTOR. Again, analogous to the last two constraints of LP-ACTOR, v^s satisfies the last two constraints of LP-ACTOR-CLASS if and only if it is a policy of the actor. The formal proof is relegated to appendix D.

LP-ACTOR-CLASS has $O(Mn^{KM})$ variables and $O(n^{2KM})$ constraints. Thus, the computation time of LP-ACTOR-CLASS is polynomial in n and exponential in K, M .

Observation 3: For $K = 2$, there exists a symmetric sensible saddle point policy of the actor $v \in \mathcal{V}^s$ such that $\sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i = 1$ if all revealed components are in state 0 and $\sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i = 0$ otherwise.

Also, when $K = 2$, for any policy of the actor, there exists a best response for the controller that is a GC policy (with a tie-break rule that may depend on the actor's policy). Using these observations, when $K = 2$, the number of variables and constraints of LP-CONTROLLER-CLASS may be reduced to $O(Mk^M)$ and $O(k^M)$ respectively.

B. Approximation guarantees using polynomial time computable policies for arbitrary systems

Saddle point policies can be computed in polynomial time when either n is a constant (using LP-CONTROLLER or LP-ACTOR) or K, M are constants (using LP-CONTROLLER-CLASS or LP-ACTOR-CLASS). The computation however becomes intractable when two or more of these parameters are large. We first develop notions of approximations for saddle-point policies. We next prove that simple linear ($O(n)$) or almost linear ($O(n \log n)$) time computable policies can provably approximate the saddle point policies as per the above notions. We also show that the approximation guarantees are tight, which in turn, completely characterize the performances of these policies. The policies we consider are intuitively appealing, and simple to implement, and hence may be of independent interest.

We first develop notions for approximations of saddle-point policies. Recall that when both players use saddle-point policies, the utility of the actor is $R_{\beta}^{u^*, v^*}$ which in turn equals the max-min and the min-max utilities of the actors. Since the actor seeks to maximize its utility, a policy of the actor may be considered a κ -approximation of its saddle-point policy, if the actor is guaranteed to obtain a utility that is at least $R_{\beta}^{u^*, v^*} / \kappa$ irrespective of the policy used by the controller. Similarly since the controller seeks to minimize the actor's utility, a policy of the controller may be considered a κ -approximation of its saddle-point policy, if this policy ensures that the actor's utility is at most $\kappa R_{\beta}^{u^*, v^*}$ irrespective of the policy used by the actor.

We show that there exists a $O(n)$ time computable $(\min(k, M) + 1)$ -approximation of the saddle-point policy for the actor. This policy, which is referred to as UA ("uniform for actor") and which is a variation of the UCA policy described earlier, selects uniformly among the concealed components and the revealed component with the highest state. Specifically, irrespective of the policy of the controller, the utility of the actor with this policy is at least $1/(\min(k, M) + 1)$ times the max-min utility of the actor for arbitrary n, K, k, M (Theorem 5.4, Section V-B.1). Thus, the worst case approximation guarantee of this policy is $(k + 1)$ (attained for large M), and the approximation guarantee when all components are statistically identical ($M = 1$) is 2. Also, the approximation improves with decrease in M and k . We also show that this approximation bound is tight in that given any M and $\epsilon > 0$, there exists a system with $K = 3$ which satisfies the following property: if the actor uses this policy, the controller can select its policy so as to upper bound the actor's utility by $1/(\min(k, M) + 1)$ times the actor's max-min utility plus ϵ (Section V-B.1). Nevertheless, our extensive simulations reveal that for large ranges of n, K, k, M, β , the minimum utility attained by the actor when he uses this policy is at least $2/3$ of the max-min utility of the actor (Section V-B.1).

We next show that there exists a $O(n \log n)$ time computable $k + 1$ -approximation of the saddle-point policy for the controller. This

policy is referred to as UGC (a GC policy that breaks ties randomly and uniformly). Specifically, irrespective of the policy of the actor, the utility of the actor when the controller uses this policy is at most $k + 1$ times the actor's min-max utility for arbitrary n, K, k, M , and at most 2 times the actor's min-max utility for arbitrary n, K, k and $M = 1$ (i.e., when all components are statistically identical) (Theorem 5.5, Section V-B.2). We also show that this approximation bound is tight in that there exists a system where $M = 2, K = 3$ and the maximum utility of the actor when the controller uses this policy is at least k times the min-max utility of the actor (Section V-B.2). Also, when $M = 1$, given any $\epsilon > 0$, there exists a system where $K = 3$ and the maximum utility of the actor when the controller uses this policy is at least $2 - \epsilon$ times the min-max utility of the actor (Section V-B.2). Again, our extensive simulations reveal that for large ranges of n, K, k, M, β , the maximum utility attained by the actor when the controller uses this policy is at most 1.3 times that of the min-max utility of the actor (Section V-B.2).

Finally, it would be useful to distinguish our notion of approximation from that of ϵ -approximate Nash equilibria [9], which bounds the loss of a player only when he unilaterally deviates (i.e., the other player uses his ϵ -approximate Nash equilibrium strategy). Our notion of approximation is therefore stronger in that it bounds the loss of utility for each player for arbitrary choices of policies by the other player. But, at the same time, owing to the notion being stronger, we are able to attain weaker guarantees in that ϵ -approximate Nash equilibria can oftentimes guarantee arbitrarily small loss, whereas we can bound the loss by specific factors, e.g., $\min(k, M) + 1, 2$ etc. Such loss bounds are oftentimes sought in the approximations algorithms literature [13].

1) *Approximation guarantees using a linear time computable policy for the actor:* Consider a symmetric sensible policy, denoted as "Uniform for Actor" or "UA", that selects each concealed class and a revealed component with equal probabilities, i.e., $UA(\mathbf{m})_i = 1/(|\mathcal{C}(\mathbf{m})| + 1)$ for each $\mathbf{m} \in \mathcal{M}, i \in \mathcal{C}(\mathbf{m})$. Note that this uniquely describes any symmetric sensible policy since a symmetric policy selects uniformly among the concealed components in each class and a sensible policy selects only a revealed component with the highest state whenever it selects a revealed component. Clearly, the actor needs $O(n)$ time and memory to select a component using this policy.

We now prove the main result of this section.

Theorem 5.4: For any β, k, n, K, M ,

$$\inf_{u \in \mathcal{U}} R_\beta^{u, \text{UA}} \geq \frac{1}{\min(k, M) + 1} \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v}.$$

Proof: Consider an arbitrary sensible policy $v \in \mathcal{V}^s$. Let $T(\mathbf{l}, \mathbf{m}, v)$ be the utility of the actor if the system state is \vec{x} such that $\mathbf{l}(\vec{x}) = \mathbf{l}$ and the actor's information is some \vec{y} such that $\mathbf{m}(\vec{y}) = \mathbf{m}$ and the actor uses the policy v . Then,

$$\begin{aligned} T(\mathbf{l}, \mathbf{m}, v) &= (1 - \sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i) R_1(\mathbf{m}) \\ &\quad + \sum_{i \in \mathcal{C}(\mathbf{m})} v^s(\mathbf{m})_i R_2(\mathbf{l}, \mathbf{m}, i) \\ &\leq \max \left(R_1(\mathbf{m}), \max_{i \in \mathcal{C}(\mathbf{m})} R_2(\mathbf{l}, \mathbf{m}, i) \right). \end{aligned} \quad (15)$$

Also,

$$\inf_{u \in \mathcal{U}} R_\beta^{u, v} = \sum_{\mathbf{l} \in \mathcal{L}} \beta''(\mathbf{l}) \min_{\mathbf{m} \in \mathcal{M}_1} T(\mathbf{l}, \mathbf{m}, v). \quad (16)$$

From (15) and (16),

$$\begin{aligned} \inf_{u \in \mathcal{U}} R_\beta^{u, \text{UA}} &= \sum_{\mathbf{l} \in \mathcal{L}} \beta(\mathbf{l}) \min_{\mathbf{m} \in \mathcal{M}_1} T(\mathbf{l}, \mathbf{m}, \text{UA}), \quad \text{where,} \quad (17) \\ T(\mathbf{l}, \mathbf{m}, \text{UA}) &= \frac{R_1(\mathbf{m}) + \sum_{i \in \mathcal{C}(\mathbf{m})} R_2(\mathbf{l}, \mathbf{m}, i)}{|\mathcal{C}(\mathbf{m})| + 1} \\ &\geq \frac{\max(R_1(\mathbf{m}), \max_{i \in \mathcal{C}(\mathbf{m})} R_2(\mathbf{l}, \mathbf{m}, i))}{\min(k, M) + 1} \quad (18) \\ &\quad (\text{since } |\mathcal{C}(\mathbf{m})| \leq \min(k, M)) \end{aligned}$$

Now, let v^* be the optimal solution of LP-ACTOR-CLASS. Then, from Theorem 5.3 and (16),

$$\sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v} = \sum_{\mathbf{l} \in \mathcal{L}} \beta''(\mathbf{l}) \min_{\mathbf{m} \in \mathcal{M}_1} T(\mathbf{l}, \mathbf{m}, v^*).$$

Thus, from (17) it is sufficient to prove that $T(\mathbf{l}, \mathbf{m}, \text{UA}) \geq T(\mathbf{l}, \mathbf{m}, v^*) / (\min(k, M) + 1)$ for each $\mathbf{l} \in \mathcal{L}, \mathbf{m} \in \mathcal{M}$.

Since v^* is sensible, the result follows from (15) and (18). ■

For $K = 2$, the approximation ratio can be improved slightly using Observation 3. It follows from Observation 3, the actor's policy that selects (a) a component in state 1 if the state of at least one such component is revealed and (b) each concealed class with equal probability, otherwise, attains a $1/\min(k, M)$ approximation ratio.

We prove that the approximation bound obtained for UA is tight. Specifically, given any $\epsilon > 0$, there exists a system with components whose state processes are mutually independent where the minimum utility obtained by the actor when it uses the uniform policy exceeds $1/(\min(k, M) + 1)$ times the max-min utility in the system by at most ϵ . Consider a system where $M > 1, K = 3$. Let the first class consist of only 1 component which is in state 2 w.p. $1 - \epsilon_1$ and in state 0 w.p. ϵ_1 . The components in the other classes are either in states 0 or 1 (the probability distributions for the state processes for channels in different classes are different). The state processes of the components are mutually independent. Let $r(2) = 1 - \delta_1, r(1) = \delta_2$. Let $v_1 \in \mathcal{V}$ be the policy that always selects the component in the first class. Clearly $R_\beta^{u, v_1} = (1 - \delta_1)(1 - \epsilon_1)$ for any $u \in \mathcal{U}$. Thus, $\sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v} \geq (1 - \delta_1)(1 - \epsilon_1)$. Consider a $u_1 \in \mathcal{U}$ that conceals the component from the first class, and selects the rest of the components to be concealed in a round robin manner. Specifically, in the first round u_1 selects one component from classes 2, \dots, M each, repeats the process in second, third rounds etc. until k components have been selected. Thus, $\min(k, M)$ classes are concealed. Clearly, the state of the component that has the highest state among the revealed components is no more than 1. Thus,

$$\begin{aligned} R_\beta^{u_1, \text{UA}} &\leq (r(2) + \min(k, M)r(1)) / (\min(k, M) + 1) \\ &\leq (1 + \min(k, M)\delta_2) / (\min(k, M) + 1) \\ &\leq \frac{(1 - \delta_1)(1 - \epsilon_1)}{(\min(k, M) + 1)} + \epsilon \quad \text{for sufficiently small } \delta_1, \delta_2, \epsilon_1 \\ &\leq \frac{\sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v}}{(\min(k, M) + 1)} + \epsilon. \end{aligned}$$

Thus, $\inf_{u \in \mathcal{U}} R_\beta^{u, \text{UA}} \leq \left(\sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v} / (\min(k, M) + 1) \right) + \epsilon$. The result follows. The scenario where this approximation factor turns out to be tight however rarely arises in practice, and as our numerical computations demonstrate, the minimum utility obtained by the uniform policy closely approximates the max-min utility of the actor in general.

We now compare, using simulations, the minimum utility attained by the actor when he uses UA with max-min utility attained by the actor. We assume $r(i) = i + 1$ throughout. We first consider the case when the states processes of all components are independent. In this case, we consider the subcases where (a) the states of each

component is selected uniformly among $0, \dots, K-1$ (b) the states of each component is selected as per a Binomial($K-1, \nu$) distribution for different ν and (c) the states of odd (even) numbered components are selected as per (a) (b). We next consider the case where the states of components are correlated. In this case, we consider the subcases where (a) only the states of first two components are correlated (i.e., if the first component is in state i , the second component is in state i w.p. α and in states adjacent to i with equal probabilities otherwise), and the states of the rest of the channel are mutually independent and (b) the states of all components are correlated (i.e., for $j > 1$ the state of component j depends on that of $j-1$ in the manner described in (a)). In each of the above subcases, we allow the state of the first component to be either fixed or distributed Uniformly or Binomially. For all these scenarios we consider different values of n, k, K such that $n \leq 6, K \leq 4, k \leq n-1$. In all of these cases, the minimum utility attained by the actor when he uses UA turns out to be at least $2/3$ of the max-min utility of the actor [12]. Thus, the performance of UA is generally significantly better than the worst case analytical bounds.

2) *Approximation guarantees using an almost linear time computable policy for the controller:* Consider UGC, the GC policy of the controller that breaks ties randomly and uniformly. Clearly, $\text{UGC} \in \mathcal{U}^s$. Note that the controller needs $O(n \log n)$ time and $O(n)$ memory to decide its actions using this policy.

Theorem 5.5: For any β, k, n, K , $\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq (k+1) \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$. For any β such that $M = 1$ and arbitrary k, n, K , $\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq 2 \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$.

Proof: The proof proceeds in three steps. The first step is to obtain a sufficiency condition for the following to hold for an arbitrary κ and arbitrary β, k, n, K, M : $\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq \kappa \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$. The next steps are to show that the above sufficiency condition is satisfied for (a) $\kappa = k+1$ and arbitrary β, k, n, K, M and (b) $\kappa = 2$ and arbitrary k, n, K and $M = 1$. The last two steps prove the two statements of the theorem respectively.

Step 1: We obtain a sufficiency condition for the following to hold for an arbitrary κ and arbitrary β, k, n, K, M : $\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq \kappa \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$. Towards this end, we will first prove that

$$\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq \kappa \inf_{u \in \mathcal{U}} R_\beta^{u, v'} \text{ for some } v' \in \mathcal{V}. \quad (19)$$

Now, $\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} \leq \kappa \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$ since $\inf_{u \in \mathcal{U}} R_\beta^{u, v'} \leq \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u, v} = \inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} R_\beta^{u, v}$.

Now, (19) can be proved as follows. Clearly,

$$\sup_{v \in \mathcal{V}} R_\beta^{\text{UGC}, v} = \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) \theta(\vec{x}) \quad (20)$$

for some real-valued function θ on \mathcal{K}^n which depends on β, k, n, K, M . Let $T'(\vec{x}, \vec{y}, v')$ be the utility of the actor if the system state is \vec{x} and the actor's information is \vec{y} and the actor uses the policy v' . Then,

$$\inf_{u \in \mathcal{U}} R_\beta^{u, v'} = \sum_{\vec{x} \in \mathcal{K}^n} \beta(\vec{x}) \min_{\vec{y} \in \mathcal{A}_c(\vec{x})} T'(\vec{x}, \vec{y}, v'). \quad (21)$$

Thus, from (20) and (21), (19) follows if we can prove that there exists a policy v' of the actor such that for each $\vec{x} \in \mathcal{K}^n$,

$$\theta(\vec{x}) \leq \kappa \min_{\vec{y} \in \mathcal{A}_c(\vec{x})} T'(\vec{x}, \vec{y}, v'). \quad (22)$$

Thus, (22) is the desired sufficiency condition.

Terminologies for Steps 2 and 3: We introduce some terminologies first. Consider an arbitrary $\vec{x} \in \mathcal{K}^n$ and $\vec{y} \in \mathcal{A}_c(\vec{x})$. Let $\text{UGC}(\vec{x})$ be the set of components whose states have been concealed by UGC when the system state is \vec{x} , $\mathcal{D}_1(\vec{x}, \vec{y}) = \text{UGC}(\vec{x}) \setminus a(\vec{y})$, and

$\mathcal{D}_2(\vec{x}, \vec{y}) = a(\vec{y}) \setminus \text{UGC}(\vec{x})$. Let \vec{x}^{UGC} be the actor's information under UGC when \vec{x} is the system state.

Note that the actor's best response to UGC is to select components whose states have been concealed since the state of any such component is at least as high as that of a component whose state has been revealed. Thus, $\theta(\vec{x}) = \sum_{i \in \text{UGC}(\vec{x})} \gamma(\vec{x}^{\text{UGC}})_i r(x_i)$ where $\gamma(\vec{x}^{\text{UGC}})$ is a probability distribution on $\text{UGC}(\vec{x})$ which depends on $\vec{x}^{\text{UGC}}, \beta, k, n, K, M$.

Step 2: We now consider the general case, that is, arbitrary β, k, n, K and construct a policy v' of the actor such that the sufficiency condition (22) holds with $\kappa = k+1$. Thus, the first statement of the theorem follows. We consider v' that selects each concealed component w.p. $1/(|a(\vec{y})|+1)$ and the revealed component with the highest state w.p. $1/(|a(\vec{y})|+1)$. Then, $T'(\vec{x}, \vec{y}, v') = (1/(|a(\vec{y})|+1)) \left(\max_{i \in \mathcal{N} \setminus a(\vec{y})} r(x_i) + \sum_{i \in a(\vec{y})} r(x_i) \right)$. Since $|a(\vec{y})| \leq k$ as $\vec{y} \in \mathcal{A}_c(\vec{x})$, (22) follows if we can show that

$$\theta(\vec{x}) \leq \max_{i \in \mathcal{N} \setminus a(\vec{y})} r(x_i) + \sum_{i \in a(\vec{y})} r(x_i) \quad \forall \vec{x} \in \mathcal{K}^n, \vec{y} \in \mathcal{A}_c(\vec{x}). \quad (23)$$

$$\begin{aligned} & \theta(\vec{x}) - \sum_{i \in a(\vec{y})} r(x_i) \\ &= \sum_{i \in \text{UGC}(\vec{x})} \gamma(\vec{x}^{\text{UGC}})_i r(x_i) - \sum_{i \in a(\vec{y})} r(x_i) \\ &\leq \sum_{i \in \mathcal{D}_1(\vec{x}, \vec{y})} \gamma(\vec{x}^{\text{UGC}})_i r(x_i) - \sum_{i \in \mathcal{D}_2(\vec{x}, \vec{y})} r(x_i) \\ &\quad (\text{since } 0 \leq \gamma(\vec{x}^{\text{UGC}})_i \leq 1 \quad \forall i \in \text{UGC}(\vec{x})) \\ &\leq \sum_{i \in \mathcal{D}_1(\vec{x}, \vec{y})} \gamma(\vec{x}^{\text{UGC}})_i r(x_i) \\ &\leq \max_{i \in \mathcal{D}_1(\vec{x}, \vec{y})} r(x_i) \quad (\text{since } \sum_{i \in \mathcal{D}_1(\vec{x}, \vec{y})} \gamma(\vec{x}^{\text{UGC}})_i \leq 1) \\ &\leq \max_{i \in \mathcal{N} \setminus a(\vec{y})} r(x_i) \quad (\text{since } \mathcal{D}_1(\vec{x}, \vec{y}) = \text{UGC}(\vec{x}) \setminus a(\vec{y}) \subseteq \mathcal{N} \setminus a(\vec{y})) \end{aligned}$$

Thus, (23) follows.

Step 3: We now consider the special case in which $M = 1$, and construct a policy v' of the actor such that the sufficiency condition (22) holds with $\kappa = 2$. Thus, the second statement of the theorem follows.

Since $M = 1$, all components are statistically identical. In this case, from symmetry, $\gamma(\vec{x}^{\text{UGC}(\vec{x})})_i = 1/k$, for each $i \in \text{UGC}(\vec{x})$, that is, the actor's best response is to select each concealed component w.p. $1/k$. Thus,

$$\theta(\vec{x}) = \sum_{i \in \text{UGC}(\vec{x})} r(x_i)/k. \quad (24)$$

We consider v' that selects (a) each concealed component w.p. $1/(2|a(\vec{y})|)$ and the revealed component with the highest state w.p. $1/2$ if at least one component is concealed and (b) the revealed component with the highest state if no component is concealed. Then,

$$T'(\vec{x}, \vec{y}, v') = \left(\max_{i \in \mathcal{N} \setminus a(\vec{y})} r(x_i) + \frac{\sum_{i \in a(\vec{y})} r(x_i)}{|a(\vec{y})|} \right) / 2.$$

Here, we assume that the second term in the sum is 0 if $a(\vec{y}) = \phi$. Thus, from (24), (22) follows if we can show that

$$\sum_{i \in \text{UGC}(\vec{x})} r(x_i)/k \leq \max_{i \in \mathcal{N} \setminus a(\vec{y})} r(x_i) + \frac{\sum_{i \in a(\vec{y})} r(x_i)}{|a(\vec{y})|} \quad (25)$$

$$\forall \vec{x} \in \mathcal{K}^n, \vec{y} \in \mathcal{A}_c(\vec{x}).$$

If $a(\vec{y}) = \phi$, the result clearly holds as then the left hand side is $\max_{i \in \mathcal{N}} r(x_i)$, and since $|\text{UGC}(\vec{x})| = k$, $\max_{i \in \mathcal{N}} r(x_i) \geq$

$\sum_{i \in \text{UGC}(\bar{x})} r(x_i)/k$. We therefore assume that $a(\bar{y}) \neq \phi$.

$$\begin{aligned}
& \sum_{i \in \text{UGC}(\bar{x})} \frac{r(x_i)}{k} - \frac{\sum_{i \in a(\bar{y})} r(x_i)}{|a(\bar{y})|} \\
\leq & \sum_{i \in \text{UGC}(\bar{x})} \frac{r(x_i)}{k} - \sum_{i \in a(\bar{y})} \frac{r(x_i)}{k} \\
& \quad (\text{since } |a(\bar{y})| \leq k \text{ as } \bar{y} \in \mathcal{A}_c(\bar{x})) \\
\leq & \sum_{i \in \mathcal{D}_1(\bar{x}, \bar{y})} \frac{r(x_i)}{k} \\
\leq & \max_{i \in \mathcal{D}_1(\bar{x}, \bar{y})} r(x_i) \\
& \quad (\text{since } |\mathcal{D}_1(\bar{x}, \bar{y})| \leq k \text{ as } \mathcal{D}_1(\bar{x}, \bar{y}) \subseteq \text{UGC}(\bar{x})) \\
\leq & \max_{i \in \mathcal{N} \setminus a(\bar{y})} r(x_i) \\
& \quad (\text{since } \mathcal{D}_1(\bar{x}, \bar{y}) = \text{UGC}(\bar{x}) \setminus a(\bar{y}) \subseteq \mathcal{N} \setminus a(\bar{y})).
\end{aligned}$$

Thus, (25) follows. \blacksquare

Note that when $K = 2$ the approximation factor turns out to be k (instead of $k + 1$) for arbitrary β, k, n, M . The proof is similar, but considers only states \bar{y} in which all revealed components are in state 0 and instead of v' considers a policy Modified Uniform for Actor or MUA $\in \mathcal{V}$ that selects (a) a revealed component that is in state 1 if the state of one such component is revealed and (b) the concealed components with equal probability if the revealed components are in state 0.

We now prove that the approximation bound obtained for UGC is tight. We first consider the case with arbitrary β , and prove that there exists a system where $M = 2$ and the maximum utility of the actor when the controller uses UGC is k times the min-max utility of the actor. Let $K = 3, r(2) = 1, r(1) = 1/k, r(0) = 0, n \geq 2k - 1$. The first class of components has k components such that one of these components is in state 2 and the rest are in state 0, and every component is as likely as any other component to be in state 2. Each component in the second class is in state 1. UGC will conceal the state of the component that is in state 2 and the states of $k - 1$ components in the second class. Consider the policy of the actor that selects a component in the first class provided one such is concealed. When the controller uses UGC, this policy always selects a component in state 1, and thus fetches the maximum possible utility, 1. Thus, the actor's maximum expected utility in this case is 1. Now, consider another policy of the controller which conceals the states of all k components in class 1 and reveals the states of the components in class 2. Now, if the actor selects a component in class 2, it attains a utility of $1/k$. If the actor selects a component in class 1, it maximizes its utility by selecting the component uniformly among the components in class 1, since it does not know the state of any component in class 1 and all components in class 1 are statistically identical. Thus, the actor's expected utility is $(1/k) \times 1 + (1 - 1/k) \times 0 = 1/k$. Thus, the actor's overall maximum expected utility is $1/k$. Thus, the min-max expected utility of the actor is at most $1/k$. Thus, the maximum utility of the actor when the controller uses UGC is at least k times the min-max utility of the actor.

We now prove that the approximation bound obtained for UGC is tight for $M = 1$. Specifically, given any $\epsilon > 0$, there exists a system with components whose state processes are identically distributed where the maximum utility obtained by the actor when the controller uses UGC exceeds $2 - \epsilon$ times the min-max utility in the system. Let $n = 2(\lceil 1/\epsilon \rceil + 1)$, $k = n/2$ and $K = 3$. Let $r(2) = 1, r(1) = 1/k, r(0) = 0$. Next, β is such that the state processes of all components are identically distributed and 1 component is in state 2, $k - 1$ components are in state 1 and the

rest of the components are in state 0³. Under UGC, the controller conceals the states of the components that are in states 1, 2. Since all components are identically distributed, when the controller uses UGC, the actor maximizes its utility by selecting uniformly and randomly among the components whose states have been concealed. Thus, the actor's expected utility is $(1/k) \times 1 + (1 - 1/k) \times (1/k) = (2/k) - (1/k^2)$. Now, consider another policy of the controller which conceals the states of (a) the component that is in state 2 and (b) $k - 1$ components that are in state 0 (selected randomly and uniformly among the components that are in state 0). Since the state processes of the components are identically distributed, in order to maximize its utility, the actor can select (a) a component whose state is revealed and which is in state 1 or (b) a component selected uniformly among those whose states have been concealed. Under (a), the actor's expected utility is $1/k$. Under (b), the actor's expected utility is $(1/k) \times 1 + (1 - 1/k) \times 0 = 1/k$. Thus, the actor's overall expected utility is $1/k$. Thus, the min-max expected utility of the actor is at most $1/k$. Note that $\frac{(2/k) - (1/k^2)}{1/k} = 2 - 1/k = 2 - 2/n > 2 - \epsilon$. Thus, the maximum utility obtained by the actor when the controller uses UGC exceeds $2 - \epsilon$ times the min-max utility in the system.

We now compare, using simulations, the maximum utility attained by the actor when the controller uses UGC with min-max utility attained by the actor. We use the same scenarios as those described in the last paragraph of Section V-B.1. When the states of the components are independent, the maximum utility attained by the actor when the controller uses UGC turns out to be very close to the min-max utility. When the states of the components are correlated, the maximum utility attained by the actor when the controller uses UGC turns out to be within 1.3 times that of the min-max utility of the actor [12]. Thus, the performance of UGC is generally significantly better than the worst case analytical bounds.

VI. GENERALIZATIONS

We have so far assumed that the controller conceals a sub-vector of the system state, and reveals the residual sub-vector. But, in general, the controller may wish to reveal a function of the system state. For example, the controller may reveal (a) limited information about each component of the system state, e.g., it may reveal for each component an interval that contains its state, or (b) a vector of a certain minimum dimensionality where each component is a function of the system state, e.g., component 1 may be the average of the first i components of the system state, etc. Next, we have assumed that the actor selects a component of the system state, and its utility is determined by the state of the component it selects. But, in general, it can select a subset of the components, and its utility may be a function of the subset it selects. For example, in cognitive radio networks, an actor may decide to transmit in more than one channels, and split its transmission power among the channels it selects; its rate of successful transmission then depends on the channels it selects and its power allocation. Our framework can be generalized to capture the above artifacts, and many of our results extend to this case.

We now describe the generalizations of the terminologies and solution concepts. For each $\bar{x} \in \mathcal{K}^n$, there exists a set $\mathcal{A}_c(\bar{x})$, such that the controller selects a member \bar{y} of $\mathcal{A}_c(\bar{x})$ as the actor's information, when the system state is \bar{x} . Here, $\mathcal{A}_c(\bar{x})$ must be designed in accordance with the constraints on the controller's actions, e.g., in previous sections $\mathcal{A}_c(\bar{x})$ consists of all sub-vectors

³This can for example be accomplished by selecting the component that is in state 2 with probability $1/n$ first, and then selecting the set of $k - 1$ components that will be in state 1 among the rest such that the probability of selecting each set of size $k - 1$ is equal, and assigning state 0 to the rest of the components.

of \vec{x} of dimension at least $n - k$, now $\mathcal{A}_c(\vec{x})$ may also consist of the range of other vector functions of \vec{x} that satisfy specific constraints, e.g., intervals containing the states of the components of \vec{x} , etc. The actor knows the vector \vec{y} selected by the controller, which may in turn reveal the controller's action (i.e., the function selected by the controller to obtain \vec{y} from \vec{x}) to the actor. When the actor's information is \vec{y} , its possible actions constitute a set $\mathcal{N}(\vec{y})$, e.g., $\mathcal{N}(\vec{y})$ may consist of possible power allocations used by the transmitter when its information is \vec{y} . If the system state is \vec{x} , and the actor selects action z , then the payoff for the actor is a function $w(\vec{x}, z)$ of both \vec{x}, z (earlier $w(\vec{x}, z)$ was $r(x_z)$). Both the controller and the actor know $n, K, \beta, \mathcal{A}_c(\vec{x})$ for each $\vec{x} \in \mathcal{K}^n$, $\mathcal{N}(\vec{y})$ for each $\vec{y} \in \mathcal{A}$, $w(\vec{x}, z)$ for each $\vec{x} \in \mathcal{K}^n$, $z \in \mathcal{N}(\vec{y})$ for each $\vec{y} \in \mathcal{A}_c(\vec{x})$. Next, a behavioral policy $u(\vec{x})$ ($v(\vec{y})$, resp.) of the controller (actor, resp.) is the probability distribution used by the controller (actor, resp.) for selecting its actions when its information is \vec{x} (\vec{y} , resp.). Specifically, $u(\vec{x})_{\vec{y}}$ ($v(\vec{y})_z$, resp.) is the probability with which the controller (actor, resp.) selects the information $\vec{y} \in \mathcal{A}_c(\vec{x})$ for the actor (selects the action $z \in \mathcal{N}(\vec{y})$, resp.) when its information is \vec{x} (\vec{y} , resp.). The controller's (actor's, resp.) utility $J_c^{u,v}(\vec{x})$ ($J_a^{u,v,\beta}(\vec{y})$, resp.) is given by $J_c^{u,v}(\vec{x}) = -E^{u,v}[r(x, B)|\vec{X} = \vec{x}]$ ($J_a^{u,v,\beta}(\vec{y}) = E_{\beta}^{u,v}[w(X, B)|\vec{Y}_a = \vec{y}]$, resp.) where B is the action of the actor. Finally, the PBE can now be described as before.

We now discuss the generalizations of the results. An equivalent zero-sum game can be obtained as in Section IV-A. Here, $R_{\beta}^{u,v}$ can be defined using $w(\vec{x}, B)$ instead of $r(x_B)$ in (3). Theorem 4.1 and Corollary 4.1 hold - the proofs remain the same. The saddle point policy for the controller can be computed using a slight modification of LP-CONTROLLER as described in p. 7. The modification is that the lower bound constraint becomes $z(\vec{y}) \geq \sum_{\vec{x}:\vec{y} \in \mathcal{A}_c(\vec{x})} \beta(\vec{x})w(\vec{x}, j)u(\vec{x})_{\vec{y}} \forall j \in \mathcal{N}(\vec{y}), \vec{y} \in \mathcal{A}_c$. Theorem 4.2 holds. The modified version of LP-CONTROLLER has $O(\sum_{\vec{x} \in \mathcal{K}^n} |\mathcal{A}_c(\vec{x})|)$ variables and $O(\sum_{\vec{y} \in \mathcal{A}_c} |\mathcal{N}(\vec{y})| + \sum_{\vec{x} \in \mathcal{K}^n} |\mathcal{A}_c(\vec{x})|)$ constraints. The saddle point policy for the actor can be computed using a slight modification of LP-ACTOR as described in p. 9. The modification is that the first constraint becomes $z(\vec{x}) \leq \sum_{i \in \mathcal{N}(\vec{y})} v(\vec{y})_i w(x, i)$ for all $\vec{y} \in \mathcal{A}_c(\vec{x}), \vec{x} \in \mathcal{K}^n$, and \mathcal{N} must be replaced by $\mathcal{N}(\vec{y})$ in the second and third constraints. Theorem 4.3 holds. The modified version of LP-ACTOR has $O(K^n + \sum_{\vec{y} \in \mathcal{A}_c} |\mathcal{N}(\vec{y})|)$ variables and $O(\sum_{\vec{x} \in \mathcal{K}^n} |\mathcal{A}_c(\vec{x})|)$ constraints. Thus, the computation times for these linear programs depend polynomially on $K^n, |\mathcal{A}_c(\vec{x})|$ for each $\vec{x} \in \mathcal{K}^n$ and $|\mathcal{N}(\vec{y})|$ for each $\vec{y} \in \mathcal{A}_c$. Next, we can generalize the UA policy for the actor - the generalization is to uniformly choose among different possible actions. We can show that for any β, n, K , $\inf_{u \in \mathcal{U}} R_{\beta}^{u, \text{UA}} \geq \frac{1}{\max_{\vec{y} \in \mathcal{A}_c} |\mathcal{N}(\vec{y})|} \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_{\beta}^{u,v}$. The proof is similar to that for Theorem 5.4. The performance guarantee may be improved if some actions in $\mathcal{N}(\vec{y})$ can be ruled out for each $\vec{y} \in \mathcal{A}_c$ for at least one saddle point policy of the actor. For example, in the special case in which $\mathcal{N}(\vec{y}) = \mathcal{N}$ for each $\vec{y} \in \mathcal{A}_c$, $|\mathcal{N}(\vec{y})| = n$, for each $\vec{y} \in \mathcal{A}_c$. But, in addition, when \mathcal{A}_c consists of the sub-vectors of the vectors in \mathcal{K}^n of size $n - k$ or more, we know that at least one saddle point policy of the actor is sensible (Corollary 4.3) and therefore selects among at most $k + 1$ possible actions irrespective of \vec{y} . Thus, the worst case approximation guarantee is $1/(k + 1)$ in Theorem 5.4. Obtaining performance guarantees for the controller, using polynomial time computation, as in Section V-B.2, however remains open.

VII. CONCLUSION

We have studied a leader-follower game where the actions of the leader (controller) determine the information available to the

follower (actor). By concealing information, the leader degrades the performance of the follower that attempts to choose one out of several resources with the best state among all. We have provided a rich body of computation and approximation tools for that problem along with mathematical foundations and complexity analysis.

REFERENCES

- [1] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(7):67–96, 1974.
- [2] R. J. Aumann and M. Maschler. *Repeated games with incomplete information*. M.I.T. Press, Cambridge, MA, 1995.
- [3] C. Daskalakis, P. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. *Commun. ACM*, 52(2):89–97, 2009.
- [4] C. Daskalakis, A. Mehta, and C. H. Papadimitriou. A note on approximate nash equilibria. *Theoretical Computer Science*, 410(17):1581–1588, 2009.
- [5] D. Fudenberg and J. Tirole. *Game Theory*. M.I.T. Press, Cambridge, MA, 2000.
- [6] A. Haurie and J. Krawczyk. *An Introduction to Dynamic Games*. <http://ecolu-info.unige.ch/~haurie/fame/textbook.pdf>.
- [7] S. Kontogiannis, P. Panagopoulou, and P. Spirakis. Polynomial algorithms for approximating nash equilibria of bimatrix games. *Theoretical Computer Science*, 410(17):1599–1606, 2009.
- [8] A. Manelli. The convergence of equilibrium strategies of approximating signaling games. *Economic Theory*, 7(2):323–335, 1996.
- [9] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, 2007.
- [10] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding — a survey. *Proceedings of the IEEE, special Issue on protection of multimedia content*, 87(7):1062–1078, July 1999.
- [11] S. Sarkar, E. Altman, R. El-Azouzi, and Y. Hayel. Information concealing games. In *Proceedings of INFOCOM mini-conference*, pages 533–539, Phoenix, AZ, April 2008.
- [12] P. Vaidyanathan. *An Empirical Investigation of Information Concealing Games in Communication Networks*. M.S. Thesis, University of Pennsylvania, Philadelphia, PA, 2007. <http://www.seas.upenn.edu/~swati/publication.htm>.
- [13] V. Vazirani. *Approximation Algorithms*. Springer, New York, 2001.

APPENDIX

A. Proof for Lemma 3.1

Proof: Recall that u^{GC} refers to an arbitrary policy in the GC class. We show that there exists a system with $n = 2, k = 1, K = 3, 0 = r(0) < r(1) < r(2) = 1$ and β under which the states of the components are mutually independent and statistically identical such that

$$\underline{R}_{\beta} < \sup_{v \in \mathcal{V}} R_{\beta}^{\text{GC}, v}. \quad (26)$$

Thus, from (6), $\inf_{u \in \mathcal{U}} R_{\beta}^{u,v} < \sup_{v \in \mathcal{V}} R_{\beta}^{\text{GC}, v}$ for each $v \in \mathcal{V}$. Thus, u^{GC} is not a saddle point policy for the controller. The lemma follows from Theorem 4.1.

Let q_i be the probability with which a component is in state i and $r(1) < q_2/(q_0 + q_2)$, $q_0 > 0, q_1 > 0$.

Let $\mathcal{U}_1^s \subset \mathcal{U}_s$ be the set of symmetric policies of the controller that conceal one component reveal the state of a component that is in state 2 only if both components are in state 2. Note that every policy in \mathcal{U}_1^s can be described by a parameter α whose role is as follows: when the system state $\vec{x} \in \{(0, 1), (1, 0)\}$, the controller reveals the component that is in state 1 with probability α . Also, $u^{\text{GC}} \in \mathcal{U}_1^s$ and corresponds to $\alpha = 0$.

Let $\mathcal{V}_1^s \subset \mathcal{V}_s$ be the set of symmetric policies of the actor that (a) never selects a revealed component that is in state 0 if a component is concealed, (b) selects a component that is in state 2 if the state of one such component is revealed and selects the component with higher state if the states of both components are revealed and one has a higher state than the other. Note that every policy in \mathcal{V}_1^s can be described by a parameter γ whose role is as follows: when a

component that is in state 1 is revealed and another component is concealed, the actor selects the revealed component with probability γ .

Using Theorem 5.1, we can prove that there exists policies $u^* \in \mathcal{U}_1^s$ and $v^* \in \mathcal{V}_1^s$ that constitute the saddle point policies of the controller and the actor respectively. For any $u' \in \mathcal{U}_1^s, v' \in \mathcal{V}_1^s$,

$$\inf_{u \in \mathcal{U}} R_\beta^{u,v'} \leq \inf_{u \in \mathcal{U}_1^s} R_\beta^{u,v'} \leq \underline{R}'_\beta \leq \overline{R}'_\beta \leq \sup_{v \in \mathcal{V}_1^s} R_\beta^{u',v} \leq \sup_{v \in \mathcal{V}} R_\beta^{u',v} \quad (27)$$

$$\begin{aligned} \text{where } \overline{R}'_\beta &= \inf_{u \in \mathcal{U}_1^s} \sup_{v \in \mathcal{V}_1^s} R_\beta^{u,v} \\ \underline{R}'_\beta &= \sup_{v \in \mathcal{V}_1^s} \inf_{u \in \mathcal{U}_1^s} R_\beta^{u,v}. \end{aligned}$$

Since $u^* \in \mathcal{U}_1^s$ and $v^* \in \mathcal{V}_1^s$ constitute the saddle point policies of the controller and the actor respectively, $\inf_{u \in \mathcal{U}} R_\beta^{u,v^*} = \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}$. Thus, all the inequalities in (27) become equalities for $u' = u^*, v' = v^*$. Thus, since $\inf_{u \in \mathcal{U}} R_\beta^{u,v^*} \leq R_\beta^{u^*,v^*} \leq \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}$, $R_\beta^{u^*,v^*} = \underline{R}'_\beta$. Also, since u^*, v^* constitute the saddle point policies of the controller and actor respectively, $R_\beta^{u^*,v^*} = \underline{R}'_\beta$. Thus, $\underline{R}'_\beta = \underline{R}'_\beta$. Also, clearly, $\sup_{v \in \mathcal{V}_1^s} R_\beta^{u^*,v} \leq \sup_{v \in \mathcal{V}} R_\beta^{u^*,v}$. Thus, (26) follows if we show that

$$\underline{R}'_\beta < \sup_{v \in \mathcal{V}_1^s} R_\beta^{u^*,v}. \quad (28)$$

Consider arbitrary $u \in \mathcal{U}_1^s, v \in \mathcal{V}_1^s$ and let α and β respectively represent u and v . First, $E_\beta^{u,v}[r(x_B)|\vec{X} = \vec{x}] = \alpha\gamma r(1) + (1-\alpha)r(1)$ if $\vec{x} \in \{(0,1), (1,0)\}$, and $E_\beta^{u,v}[r(x_B)|\vec{X} = \vec{x}] = \gamma r(1) + (1-\gamma)$ if $\vec{x} \in \{(1,2), (2,1)\}$. Next, $E_\beta^{u,v}[r(x_B)|\vec{X} = \vec{x}]$ does not depend on α, γ if $\vec{x} \notin \{(0,1), (1,0), (1,2), (2,1)\}$. Also, $\beta(\vec{x}) = q_0 q_1$ if $\vec{x} \in \{(0,1), (1,0)\}$, and $\beta(\vec{x}) = q_1 q_2$ if $\vec{x} \in \{(1,2), (2,1)\}$. Thus, from (3),

$$\begin{aligned} R_\beta^{u,v} &= 2q_0 q_1 (\alpha\gamma r(1) + (1-\alpha)r(1)) \\ &\quad + 2q_1 q_2 (\gamma r(1) + (1-\gamma)) + C, \end{aligned} \quad (29)$$

where C is a constant that depends on q_0, q_1, q_2 but not α, γ .

Since $\alpha = 0$ for u^{GC} , from (29), for $v \in \mathcal{V}_1^s$, $R_\beta^{u^{\text{GC}},v} = 2q_0 q_1 r(1) + 2q_1 q_2 (\gamma r(1) + (1-\gamma)) + C$. Thus,

$$\begin{aligned} \sup_{v \in \mathcal{V}_1^s} R_\beta^{u^{\text{GC}},v} &= 2q_0 q_1 r(1) + C \\ &\quad + 2q_1 q_2 \max_{0 \leq \gamma \leq 1} (\gamma r(1) + (1-\gamma)) \\ &= 2q_0 q_1 r(1) + C \\ &\quad + 2q_1 q_2 \quad (\text{since } r(1) < 1). \end{aligned} \quad (30)$$

Now, from (29),

$$\begin{aligned} \underline{R}'_\beta &= C + \max_{0 \leq \gamma \leq 1} \min_{0 \leq \alpha \leq 1} 2q_0 q_1 (\alpha\gamma r(1) + (1-\alpha)r(1)) \\ &\quad + 2q_1 q_2 (\gamma r(1) + (1-\gamma)) \\ &= C + 2q_1 \max_{0 \leq \gamma \leq 1} (\gamma r(1)(q_0 + q_2) + (1-\gamma)q_2) \\ &\quad (\text{since } \gamma \leq 1, r(1) \geq 0) \\ &= C + 2q_1 q_2 \text{ since } r(1) < q_2/(q_0 + q_2). \end{aligned}$$

Now, (28) follows from (30) since $q_0 > 0, q_1 > 0, r(1) > 0$. ■

B. Proof for Lemma 3.2

Proof: Let v^{SBA} refer to an arbitrary policy in the SBA class. Recall the description of the MUA policy for the actor at the end of Section V-B.1. We show that there exists a system with $n = 3, k = 2, K = 2, r(0) = 0, r(1) = 1, \beta$ under which the states of the

components are mutually independent, a policy u' of the controller such that

$$R_\beta^{u',v^{\text{SBA}}} < \inf_{u \in \mathcal{U}} R_\beta^{u,\text{MUA}}. \quad (31)$$

Thus, since $\inf_{u \in \mathcal{U}} R_\beta^{u,v^{\text{SBA}}} \leq R_\beta^{u',v^{\text{SBA}}}$, and $\inf_{u \in \mathcal{U}} R_\beta^{u,\text{MUA}} \leq \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} R_\beta^{u,v} = \underline{R}'_\beta$, $\inf_{u \in \mathcal{U}} R_\beta^{u',v^{\text{SBA}}} < \underline{R}'_\beta$. Thus, from (6), $\inf_{u \in \mathcal{U}} R_\beta^{u',v^{\text{SBA}}} < \sup_{v \in \mathcal{V}} R_\beta^{u',v}$ for each $u \in \mathcal{U}$. Thus, v^{SBA} is not a saddle point policy for the actor. The lemma follows from Theorem 4.1.

Let q_i be the probability with which a component is in state i and $q_1 > \max(q_2, q_3)$. Thus, v^{SBA} selects component 1 whenever component 1 has been concealed and no revealed component is in state 1.

Let $u' \in \mathcal{U}$ (a) conceal 2 components and never reveal a component that is in state 1 unless all components are in state 1 and (b) conceal component 1 unless both components 2 and 3 are in state 1, and reveal component 1 otherwise. Now,

$$R_\beta^{u',v^{\text{SBA}}} = q_1 + (1-q_1)q_2 q_3. \quad (32)$$

Clearly, for any $u \in \mathcal{U}$, $R_\beta^{u,\text{MUA}} \geq \Theta_1/2 + \Theta_2$, where Θ_1 is the probability that only one component is in state 1 and Θ_2 is the probability that two or more components are in state 1. Now, $\Theta_1 = q_1(1-q_2)(1-q_3) + q_2(1-q_1)(1-q_3) + q_3(1-q_1)(1-q_2)$ and $\Theta_2 = q_1(1-(1-q_2)(1-q_3)) + (1-q_1)q_2 q_3$. Thus, $\Theta_1/2 + \Theta_2 = q_1 + (1-q_1)q_2 q_3 - q_1(1-q_2)(1-q_3)/2 + q_2(1-q_1)(1-q_3)/2 + q_3(1-q_1)(1-q_2)/2$. We now show that there exists $q_1 > q_2 > q_3$ such that $q_2(1-q_1)(1-q_3) + q_3(1-q_1)(1-q_2) - q_1(1-q_2)(1-q_3) > 0.0625$. Thus, for any $u \in \mathcal{U}$, $R_\beta^{u,\text{MUA}} \geq \Theta_1/2 + \Theta_2 > q_1 + (1-q_1)q_2 q_3 + 0.0625$. Hence, (31) follows from (32).

Let $q_1 = 0.5, q_2 = 0.5 - \epsilon_1, q_3 = 0.5 - \epsilon_2$, where $\epsilon_1 > 0$ and $\epsilon_2 > 0$. Note that $q_2(1-q_1)(1-q_3) + q_3(1-q_1)(1-q_2) = (1-q_1)(q_2 + q_3 - 2q_2 q_3) = 0.5(1-\epsilon_1-\epsilon_2-2(0.5-\epsilon_1)(0.5-\epsilon_2))$. Next, $q_1(1-q_2)(1-q_3) = 0.5(1-q_2-q_3+q_2 q_3) = 0.5(\epsilon_1 + \epsilon_2 + (0.5-\epsilon_1)(0.5-\epsilon_2))$. Thus,

$$\begin{aligned} & q_2(1-q_1)(1-q_3) + q_3(1-q_1)(1-q_2) - q_1(1-q_2)(1-q_3) \\ &= 0.5(1-2\epsilon_1-2\epsilon_2-3(0.5-\epsilon_1)(0.5-\epsilon_2)) \\ &> 0.0625 \text{ for small } \epsilon_1, \epsilon_2. \end{aligned}$$

■

C. Proof for Theorem 5.2

Proof: Consider the description of LP-CONTROLLER at the end of Section IV-B.1, and restrict the feasible solutions u to \mathcal{U}^s . From Theorem 5.1, the optimal solution of LP-CONTROLLER is a saddle point policy of the controller even with this restriction, and the optimal solution is clearly a symmetric policy for the controller. The last two constraints in LP-CONTROLLER-CLASS ensure that its optimal solution is a symmetric policy of the controller. Thus, we only need to show that there is a one-to-one correspondence between the sets of optimal solutions of LP-CONTROLLER-CLASS and LP-CONTROLLER with the above restriction, such that the corresponding solutions in the two sets provide the same policies.

Consider LP-CONTROLLER with the additional constraint that $u \in \mathcal{U}^s$. Let $\mathcal{L}(\vec{y}) = \{1 : 1(\vec{x}) = 1 \text{ for some } \vec{x} \text{ s.t. } \vec{y} \in \mathcal{A}_c(\vec{x})\}$. Note that $\mathcal{L}(\vec{y})$ depends on \vec{y} only through $\mathbf{m}(\vec{y})$, and can therefore be denoted as $\mathcal{L}(\mathbf{m}(\vec{y}))$. Since $u \in \mathcal{U}^s$, for each $\vec{y} \in \mathcal{A}_{c,k}$, we can write the first constraint as

$$z(\vec{y}) \geq R_1(\mathbf{m}(\vec{y})) \sum_{1 \in \mathcal{L}(\mathbf{m}(\vec{y}))} \beta'(1) u'(1)_{\mathbf{m}(\vec{y})} \Theta_2(1, \mathbf{m}(\vec{y})). \quad (33)$$

