

# Verification of Nash-Equilibrium for Probabilistic BAR Systems

Fernando Dileepa\*, Naipeng Dong<sup>†</sup>, Cyrille Jegourel<sup>‡</sup>, Jin Song Dong<sup>§</sup>

National University of Singapore

{fdileepa\*, dongnp<sup>†</sup>, jegourel<sup>‡</sup>, dongjs<sup>§</sup>}@comp.nus.edu.sg

**Abstract**—A BAR system specifies a cooperation between agents who can be altruistic when they follow the specified behaviours, Byzantine when they randomly deviate from specifications and rational when they deviate to increase their own benefits. We consider whether a rational agent indeed follows the specification of a probabilistic BAR system as verifying whether the system is a Nash-equilibrium in the corresponding stochastic games. In this article, we propose an intuitive specification for probabilistic BAR systems and an algorithm to automatically verify Nash-equilibrium. To validate our implementation of the algorithm, we present two case studies – the three-player Rock-paper-scissors game and a probabilistic secret sharing protocol.

## I. INTRODUCTION

Game Theory is the study of mathematical models of conflict and cooperation between multiple agents who are considered as intelligent and rational decision-makers, and is used in a large range of domains (economics, psychology, computer science...). In communication networks, the protocols are seen as games in which the involved entities of the network must cooperatively follow the protocol rules. However, it may be possible that a particular entity is corrupted or does not work correctly with respect to the protocol. Some device may have been designed to deviate from the protocol to maximise a self-interest function or may be simply misconfigured. A protocol is unlikely to work correctly with respect to a property in the presence of selfish or “broken” agents. However, it remains of first importance to guarantee some safety property under these conditions. For example, in the secret sharing protocol, we expect that a corrupted agent cannot reconstruct highly sensitive information before the other agents. Byzantine-Altruistic-Rational (BAR) models have been introduced in [1] in order to analyse the protocol correctness of systems under these conditions. In these systems, the agents are divided in three categories of players. They are said to be altruistic if they follow the rules of the protocol, even if a rational choice is to deviate from it. The agents are rational if they may deviate from the protocol in order to maximise their self-interest, captured in a utility function. Finally, Byzantine agents characterise the players who arbitrarily deviate from the protocol because of component failures, malicious intent, security compromise. The general architecture of BAR systems have been introduced in [1]. In the BAR framework, the correctness of the protocol with respect to some property is satisfied if the property remains true in spite of the presence of rational and Byzantine players. Such protocols are said

BAR-tolerant. Some of them have been already proposed to implement cooperative services in peer-to-peer data streaming applications [2].

BAR-tolerant protocols are thus important because, they guarantee that selfish agents actually play as expected by the protocol even if some agents are identified as broken or misconfigured. However, it remains challenging to prove that a protocol is BAR-tolerant. For that purpose, it is relevant to make use of Nash-equilibriums, a common game theory concept characterising strategies from which rational players should not deviate in order to maximise their utility. In [3], the authors presented a symbolic Model Checking algorithm that automatically verifies whether the proposed protocol is a *Nash-equilibrium*. In [4], PRALINE has been presented as a tool for computing Nash-equilibrium in non-probabilistic concurrent games played on graphs. By concurrent games, we mean games where the next state is defined at the beginning of a turn by the individual choices of all the players. However, these previous works are limited to non-*probabilistic protocols*. Moreover, the game must be modelled with perfect information, meaning that the rational agents are assumed to have a perfect knowledge of the player utilities and of the global state of the game.

In our article, we focus on a particular class of probabilistic protocols, that is protocols in which the agents take decisions accordingly to a specified probability distribution. The altruistic agents play thus with respect to this distribution, the rational agents play with respect to the probability distribution that maximises their expected utility and the Byzantine agents play accordingly to any probability distribution. Note that we do not cover the case in which illegal decisions with respect to the BAR system may be taken by the rational and Byzantine players. Moreover, we consider probabilistic games in which the utilities are known by the rational agents, as well as the number of Byzantine players.

Related to our work, the tool PRISM-games [5] computes an optimal strategy for a coalition of rational players in a probabilistic game. This strategy can then be used to verify that the global strategy is a Nash-equilibrium. However, PRISM-games is limited to the analysis of turn-based games, that are games where agents select their moves in turns. Synchronisation between agents is thus not available directly in PRISM-games. Moreover, the protocols are with perfect information: the agents know the whole reward structure as well as the current global state of the game and the probability distributions of

all the players. Finally, PRISM-games computes an optimal strategy up to some bounded number of iterations but it does not compute explicitly a potentially larger bound guaranteeing that a particular strategy is a Nash-equilibrium.

#### A. Contributions and structure of the article

In Section II, we present the specification of parallel and synchronous probabilistic BAR systems. In Section III, we propose a verification algorithm to verify Nash-equilibrium of the system. Finally, we apply the algorithm in two case studies, one with perfect information and one with imperfect information in Section V and we present interesting results with regard of Byzantine agents. We summarise our results in Section VIII. For the best of our knowledge, this is the first attempt of verifying Nash-equilibrium of probabilistic BAR systems. Nash-equilibrium is verified for imperfect information scenario as well as the perfect information scenario. This differs from previous works e.g., [5] and [3], since we consider imperfect information games and since we verify Nash-equilibriums explicitly instead of approximate equilibriums [5]. Finally, our verification algorithm determines a bound for the number of iterations that guarantees the Nash-equilibrium property.

## II. SPECIFICATION

In BAR systems, the non-altruistic agents do not necessarily follow the protocol rules. It is thus necessary to capture their possible deviations through adequate specifications. Moreover, we consider probabilistic BAR systems. In this setting, the agents may have to choose a local action among several enabled actions. Their choices are governed by a probability distribution. In this section, we present a framework which allows us to formally specify a probabilistic BAR system as a game in an intuitive way. We highlight that our framework facilitates modelling of concurrent games with some imperfect information. In our case, we restrict the imperfect information for a player to the knowledge of her opponents' local states. It means that each agent knows her own local state, the entire reward structure of the game, as well as, given a global state, the outgoing probability distribution.

We first show the formal modelling of each type of players as Finite State Machines (FSM) and then depict the combination of the FSMs to form a global transition system modelling the global behaviour of the whole system.

#### A. Altruistic players

Altruistic player's behaviours that are specified in a protocol, can be formally captured by a FSM defined as a tuple  $\mathcal{M}_i^a = (S_i^a, I_i^a, A_i^a, G_i^a, T_i^a, P_i^a, H_i^a)$  where

- $S_i^a$  is the (finite) set of states of altruistic player  $i$ ,
- $I_i^a$  denotes the set of initial states,
- $A_i^a$  is the set of actions,
- $G_i^a$  is the set of atomic propositions defined on global states  $S$  that serves as guard conditions of taking an action and gaining rewards,
- $T_i^a : [\phi]S_i^a \times A_i^a \rightarrow S_i^a$ , with  $\phi$  being a first-order logic formula over  $G_i^a$ , specifies the transitions between states

and the condition  $\phi$  that needs to be satisfied for taking this transition,

- $P_i^a : S_i^a \times A_i^a \rightarrow [0, 1]$  such that  $\forall s_i \in S_i^a, \sum_{a_i \in A_i^a} P_i^a(s_i, a_i) = 1$  specifies the probabilities of taking an action at state  $s_i$ ,
- $H_i^a$  has two functions  $AH_i^a : S_i^a \times A_i^a \rightarrow \mathbb{R}$  specifies the pay-off of an action  $a_i \in A_i^a$  at state  $s_i \in S_i^a$ , and  $SH_i^a : [\phi]S \rightarrow \mathbb{R}$  specifies the pay-off obtained after reaching states  $S$  satisfying formula  $\phi$ .

When the system is well-defined, altruistic players have either a probabilistic or a deterministic behaviour, that is, at state  $s_i \in S_i^a$ , there is either one action  $a_i \in A_i^a$  or a set of probabilistic actions whose probabilities sum up to 1. When the action is deterministic, the probability is omitted, since the probability is always 1.

**Example 1. A three-player Rock-paper-scissors game** Consider a 3-player machine-game version of Rock-paper-scissors derived from [6]. We model playing rock, paper, and scissors as playing 0, 1, and 2, respectively. We denote  $(i, j, k)$  as the respective outcomes of players  $p_1, p_2$  and  $p_3$ , denote  $+$  as the operator for the addition modulo 3 and denote  $a = b \oplus c$  as  $a = b$  or, exclusively,  $a = c$ . The pay-off of player  $p_1$  for the outcome  $(i, j, k)$  is 2 if  $j = k$  and  $i = j + 1$  ( $p_1$  wins uniquely), 1 if  $i = k \oplus j$  and  $i - 1 = (j \oplus k)$  (both  $p_1$  and another player win),  $-1$  if  $i + 1 = j \oplus k$  ( $p_1$  loses to one player),  $-2$  if  $j = k$  and  $j = i + 1$  ( $p_1$  loses to both players), and 0 if  $i = j = k$  or  $i + 1 = j \oplus k \wedge i - 1 = k \oplus j$  (no one wins and three players play the same, or  $p_1$  wins one player but loses to another). Player  $p_2$ 's pay-off is similar to the pay-off of player  $p_1$  after replacing  $i, j, k$  by  $j, k, i$  in player  $p_2$ 's pay-off conditions. Player  $p_3$ 's pay-off is the negatives of player  $p_1$ 's plus player  $p_2$ 's pay-off.

We model altruistic player's behaviour in Example 1 as follows.

- $S_i^a = \{s_1^a, s_2^a, s_3^a, s_4^a\}$ ,  $I_i^a = \{s_1^a\}$ ,  $A_i^a = \{0, 1, 2, \text{replay}\}$ ,
- $G_i^a = \{i = j, j = k, i = j + 1, j = i + 1, i = k \oplus j, i - 1 = (j \oplus k), i + 1 = j \oplus k\}$ ,
- $T_i^a(s_1^a, 0) = s_2^a$ ,  $T_i^a(s_1^a, 1) = s_3^a$ ,  $T_i^a(s_1^a, 2) = s_4^a$ ,  $[\phi]T_i^a(s_4^a, \text{replay}) = s_1^a$ ,  $[\phi]T_i^a(s_3^a, \text{replay}) = s_1^a$ ,  $[\phi]T_i^a(s_2^a, \text{replay}) = s_1^a$ , with  $\phi = \neg((j = k \wedge i = j + 1) \vee (i = k \wedge j = i + 1) \vee (i = j \wedge k = i + 1))$ , meaning that the game terminates when there is a unique winner, and continues otherwise,
- $P_i^a(s_1^a, 0) = 1/3$ ,  $P_i^a(s_1^a, 1) = 1/3$ ,  $P_i^a(s_1^a, 2) = 1/3$ , and action *replay* is deterministic,
- $SH_i^a(S) = \begin{cases} 2 & \text{if } S \models (j = k \wedge i = j + 1) \\ 1 & \text{if } S \models (i = k \oplus j \wedge i - 1 = (j \oplus k)) \\ 0 & \text{if } S \models ((i = j = k) \vee (i + 1 = j \oplus k \wedge i - 1 = k \oplus j)) \\ -1 & \text{if } S \models (i + 1 = j \oplus k) \\ -2 & \text{if } S \models (j = k \wedge j = i + 1). \end{cases}$

### B. Byzantine players

Byzantine players may randomly deviate from the protocol by 1) performing actions that are not specified in the protocol, or 2) performing a set of probabilistic actions with a different probabilistic distribution. The deviating actions at a state shall be specified by the BAR system. As we want to ensure the BAR-tolerance of protocols, we only consider the worst-case scenario for the rational players, meaning that we assume that the Byzantine players always perform an action that minimizes the pay-off of the rational player. This action is unique, for that any distribution over the actions would not minimise the rational player's pay-off larger than one single action. Hence, there is no probability distribution needed for Byzantine players. In addition, since Byzantine players randomly deviate, the guard conditions and pay-off function are not needed neither.

Similar to the altruistic player, a Byzantine player can be formally modelled by a FSM, a tuple  $\mathcal{M}_i^b = (S_i^b, I_i^b, A_i^b, T_i^b)$ , where

- $S_i^b$  is the (finite) set of states of Byzantine player  $i$ ,
- $I_i^b$  denotes the set of initial states,
- $A_i^b$  is the set of actions,
- $T_i^b : S_i^b \times A_i^b \rightarrow S_i^b$  specifies the transitions between states.

Assume that the Byzantine players in Example 1 can only deviate from the specification by playing according to a different probability distribution than the uniform distribution. A Byzantine player is modelled as follows:

- $S_i^b = \{s_1^b, s_2^b, s_3^b, s_4^b\}$ ,  $I_i^b = \{s_1^b\}$ ,  $A_i^b = \{0, 1, 2, replay\}$ ,
- $T_i^b(s_1^b, 0) = s_2^b$ ,  $T_i^b(s_1^b, 1) = s_3^b$ ,  $T_i^b(s_1^b, 2) = s_4^b$ ,  
 $T_i^b(s_4^b, replay) = s_1^b$ ,  $T_i^b(s_3^b, replay) = s_1^b$ ,  
 $T_i^b(s_2^b, replay) = s_1^b$ .

### C. Rational players

Similar to Byzantine players, the rational players may also deviate from the protocol specification. Unlike Byzantine players, whose actions are random, the rational players' actions are driven by pay-offs, that is, at a state, the rational players choose the action that leads to larger pay-off. Similarly to the Byzantine players, this action is unique. In fact, a rational player has the same allowed transitions, states and actions as a Byzantine player. Unlike to the Byzantine players, the rational players need the pay-off function. Hence, a rational player can be formally modelled by a FSM, a tuple  $\mathcal{M}_i^r = (S_i^r, I_i^r, A_i^r, G_i^r, T_i^r, H_i^r)$ , where

- $S_i^r$  is the (finite) set of states of rational player  $i$ ,
- $I_i^r$  denotes the set of initial states,
- $A_i^r$  is the set of actions,
- $G_i^r$  is the set of atomic propositions defined on global states  $S$  that serves as guard conditions of gaining rewards,
- $T_i^r : S_i^r \times A_i^r \rightarrow S_i^r$  specifies the transitions between states,

- $H_i^r$  has two functions:  $AH_i^r : S_i^r \times A_i^r \rightarrow \mathbb{R}$  specifies the pay-off of an action  $a_i \in A_i^r$  at state  $s_i \in S_i^r$ , and  $SH_i^r : [\phi]S \rightarrow \mathbb{R}$  specifies the pay-off of reaching states  $S$  satisfying formula  $\phi$ .

A rational player in Example 1 is modelled as follows:

- $S_i^r = \{s_1^r, s_2^r, s_3^r, s_4^r\}$ ,  $I_i^r = \{s_1^r\}$ ,  $A_i^r = \{0, 1, 2, replay\}$ ,
- $T_i^r(s_1^r, 0) = s_2^r$ ,  $T_i^r(s_1^r, 1) = s_3^r$ ,  
 $T_i^r(s_1^r, 2) = s_4^r$ ,  $T_i^r(s_4^r, replay) = s_1^r$ ,  
 $T_i^r(s_3^r, replay) = s_1^r$ ,  $T_i^r(s_2^r, replay) = s_1^r$ ,
- $G_i^r = G_i^a$ ,
- $SH_i^r(S) = \begin{cases} 2 & \text{if } S \models (j = k \wedge i = j + 1) \\ 1 & \text{if } S \models (i = k \oplus j \wedge i - 1 = (j \oplus k)) \\ 0 & \text{if } S \models (i = j = k) \vee \\ & (i + 1 = j \oplus k \wedge i - 1 = k \oplus j) \\ -1 & \text{if } S \models (i + 1 = j \oplus k) \\ -2 & \text{if } S \models (j = k \wedge j = i + 1). \end{cases}$

### D. Global graph

Given a finite set of players modelled as FSMs  $(\{\mathcal{M}_1, \dots, \mathcal{M}_n\})$ , one can construct a global FSM  $\mathcal{M} = (S, I, A, G, T, P, H)$  modelling the concurrent execution of player's FSMs. Note that in our algorithm, the global graph is constructed implicitly on-the-fly. For the clearer presentation of our verification in the next section, we explicitly construct a global graph here to provide an intuition on how a global graph is constructed.

The global states in the global FSM is the product of the local states of each player, i.e.,  $S = S_1 \times S_2 \times \dots \times S_n$ ; the set of initial states is  $I = I_1 \times I_2 \times \dots \times I_n$ ; the set of global actions is  $A = A_1 \times A_2 \times \dots \times A_n$ ; the set of atomic propositions is  $G = G_1 \cup G_2 \cup \dots \cup G_n$  ( $G_i$  is empty if  $\mathcal{M}_i$  models a Byzantine player).

Intuitively, the transition function  $T$  is defined as  $T(\langle s_1, \dots, s_n \rangle, \langle a_1, \dots, a_n \rangle) = \langle s'_1, \dots, s'_n \rangle$  where  $T_i(s_i, a_i) = s'_i$  in  $\mathcal{M}_i$  meaning that at state  $\langle s_1, \dots, s_n \rangle$  every player takes an action according to their local FSMs, then there is a transition in the global FSM. Note that we require that every player has to take an action: Altruistic players always follow the protocol specification to take an action; Byzantine and rational players may deviate from the specification by doing nothing or taking other actions, where the deviation actions, including doing nothing, shall be defined and modelled in the local FSMs. We emphasize that we model each player's local actions rather than requiring the global FSM as input like in [3]. Hence, we need to additionally consider the special cases where multiple users need to take actions at the same time, i.e., action synchronization, e.g., in Example 1, three players need to take the *replay* action at the same time, in order to re-start the game. We support two ways of modelling synchronized actions: one is to model action as one name with an arity, (e.g. the number of players), such that more than the arity number of players taking the action can be synchronized, e.g., the *replay/3* means that at least three players want to re-start the game; the other is to define a pair of actions, for example (*send*, *receive*) in communication, meaning that whenever there is a *send* action

and a *receive* action are enabled at a state, the synchronized action can be taken. The synchronized actions need to be tagged in a set  $A_s$ . Given a global action  $\langle a_1, \dots, a_n \rangle$ , let  $J(a)$  be defined as

$\forall a/arity \in A_s, J(a) \subset \{1, \dots, n\}$  s.t.  $\forall j \in J(a) a_j = a$  with  $|J(a)| = arity$ .  $J(a)$  specifies the exact set of positions that the synchronised action  $a/arity$  can be activated.

In Example 1,  $A_s = \{replay/3\}$ . When an action  $\langle a_1, \dots, a_n \rangle$  is enabled at a state  $s_i$ , if  $a_i$  is in  $A_s$ , but there is no other action to synchronize with, for the simplicity of modelling, we assume a failure state  $s_f$  exists and  $T(s, \langle a_1, \dots, a_n \rangle) = s_f$ . Formally,  $T(s, \langle a_1, \dots, a_n \rangle) = s_f$ , if  $\exists a/arity \in A_s$  s.t.  $0 < |J(a)| < n$ . where  $|A|$  denotes the number of elements in set  $A$ .

Generally, the probabilistic function is  $P(\langle s_1, \dots, s_n \rangle, \langle a_1, \dots, a_n \rangle) = P_1(s_1, a_1) \times \dots \times P_n(s_n, a_n)$ . If  $P_i(s_i, a_i)$  is not defined i.e., in the case of deterministic or non-deterministic actions, we set the probability value to be 1. It is straightforward in the case that  $a_i$  is a deterministic action. In the case of non-deterministic actions, once  $a_i$  is chosen, it becomes deterministic, and thus the value is also 1. In such a way, the probability of the global action  $\langle a_1, \dots, a_n \rangle$  at state  $\langle s_1, \dots, s_n \rangle$  i.e.,  $\prod_{i=1}^n P_i(s_i, a_i)$ , actually captures the product of the probabilities of actions in  $\{a_1, \dots, a_n\}$ , and thus the probabilities of all global actions at a state sum up to 1. When no action is probabilistic in the global action  $\langle a_1, \dots, a_n \rangle$ , then we do not sign a probability to the action at the state, denoted as  $P(\langle s_1, \dots, s_n \rangle, \langle a_1, \dots, a_n \rangle) = \circ$ , meaning the probability is not defined.

The pay-off function  $AH$  is defined as  $H(s, \langle a_1, \dots, a_n \rangle) = \langle H_1(s, a_1), \dots, H_n(s, a_n) \rangle$  where  $H_i(s, a_i)$  can be empty, denoted as  $\star$ , meaning that function  $H_i$  is not defined for action  $a_i$  at state  $s$ . Similarly,  $SH$  can be defined as,  $H(\langle s_1, \dots, s_n \rangle) = \langle H_1(s_1), \dots, H_n(s_n) \rangle$ .

Given one altruistic player, one Byzantine player, one rational player, the global FSM for Example 1 is constructed using the above way. Since there are too many states in the global FSM, we only demonstrate some interesting parts of the global FSM, for example, we only show the states with incoming and outgoing transitions.

- $S = \{ \langle s_1^a, s_1^b, s_1^r \rangle, \langle s_2^a, s_2^b, s_2^r \rangle, \langle s_3^a, s_3^b, s_3^r \rangle, \langle s_2^a, s_3^b, s_2^r \rangle, \langle s_2^a, s_2^b, s_3^r \rangle, \langle s_3^a, s_3^b, s_2^r \rangle, \langle s_3^a, s_2^b, s_3^r \rangle, \langle s_2^a, s_3^b, s_4^r \rangle, \langle s_4^a, s_4^b, s_2^r \rangle, \langle s_4^a, s_2^b, s_4^r \rangle, \langle s_2^a, s_4^b, s_4^r \rangle, \langle s_3^a, s_4^b, s_4^r \rangle, \langle s_4^a, s_3^b, s_4^r \rangle, \langle s_4^a, s_4^b, s_3^r \rangle, \langle s_3^a, s_3^b, s_4^r \rangle, \langle s_3^a, s_4^b, s_3^r \rangle, \langle s_4^a, s_3^b, s_3^r \rangle \}$ ,
- $I = \{ \langle s_1^a, s_1^b, s_1^r \rangle \}$ ,
- $A = \{ \langle a_1, a_2, a_3 \rangle | a_1, a_2, a_3 \in \{0, 1, 2, replay\} \text{ and } A_s = \{replay/3\} \}$ ,
- $G = G_i^a$ ,
- The playing transitions are:
  - $T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 0, 0 \rangle) = \langle s_2^a, s_2^b, s_2^r \rangle$ ,
  - $T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 0, 0 \rangle) = \langle s_3^a, s_2^b, s_2^r \rangle$ ,
  - $T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 1, 0 \rangle) = \langle s_2^a, s_3^b, s_2^r \rangle$ ,
  - $T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 0, 1 \rangle) = \langle s_2^a, s_2^b, s_3^r \rangle$ ,

$$\begin{aligned}
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 1, 0 \rangle) &= \langle s_3^a, s_3^b, s_2^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 0, 1 \rangle) &= \langle s_3^a, s_2^b, s_3^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 1, 1 \rangle) &= \langle s_2^a, s_3^b, s_3^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 1, 1 \rangle) &= \langle s_3^a, s_3^b, s_3^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 0, 0 \rangle) &= \langle s_4^a, s_2^b, s_2^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 2, 0 \rangle) &= \langle s_2^a, s_4^b, s_2^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 0, 2 \rangle) &= \langle s_2^a, s_2^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 2, 0 \rangle) &= \langle s_4^a, s_4^b, s_2^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 0, 2 \rangle) &= \langle s_4^a, s_2^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, 2, 2 \rangle) &= \langle s_2^a, s_4^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 2, 2 \rangle) &= \langle s_4^a, s_4^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 2, 2 \rangle) &= \langle s_3^a, s_4^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 1, 2 \rangle) &= \langle s_2^a, s_3^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 2, 1 \rangle) &= \langle s_4^a, s_4^b, s_3^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 1, 2 \rangle) &= \langle s_3^a, s_3^b, s_4^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, 2, 1 \rangle) &= \langle s_3^a, s_4^b, s_3^r \rangle, \\
 T(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, 1, 1 \rangle) &= \langle s_4^a, s_3^b, s_3^r \rangle,
 \end{aligned}$$

- and the re-start transitions are:

$$\begin{aligned}
 T(\langle s_2^a, s_2^b, s_2^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_3^a, s_3^b, s_2^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_3^a, s_2^b, s_3^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_2^a, s_3^b, s_3^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_3^a, s_3^b, s_3^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_2^a, s_2^b, s_2^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_2^a, s_4^b, s_2^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_2^a, s_2^b, s_4^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_4^a, s_4^b, s_4^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_3^a, s_4^b, s_4^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_4^a, s_3^b, s_4^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle, \\
 T(\langle s_4^a, s_4^b, s_3^r \rangle, \langle replay, replay, replay \rangle) &= \langle s_1^a, s_1^b, s_1^r \rangle.
 \end{aligned}$$

The re-start transitions only happen when there is no unique winner. When there is a unique winner, the game terminates. In this case, although the Byzantine player's and the rational player's *replay* actions are enabled, there will not be a global transition, because the rational player will not take an action.

- $P(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 0, a_i, a_j \rangle) = 1/3 \times 1 \times 1 = 1/3$  (the Byzantine and rational player actions  $a_i, a_j$  are non-deterministic, and thus  $P^b(s_1^b, a_i) = 1, P^r(s_1^r, a_j) = 1$ ). Similarly,  $P(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 1, a_i, a_j \rangle) = 1/3$ ,  $P(\langle s_1^a, s_1^b, s_1^r \rangle, \langle 2, a_i, a_j \rangle) = 1/3$ . In addition,  $P(\langle s_1^a, s_1^b, s_1^r \rangle, \langle replay, replay, replay \rangle) = \circ$
- $SH(s, \langle a_1, a_2, a_3 \rangle) =$ 

$$\begin{cases}
 \langle 2, -1, \star \rangle & \text{if } S \models (j = k \wedge i = j + 1) \\
 \langle -1, 2, \star \rangle & \text{if } S \models (i = k \wedge i + 1 = j) \\
 \langle -1, -1, \star \rangle & \text{if } S \models (i = j \wedge i + 1 = k) \\
 \langle 0, 0, \star \rangle & \text{if } S \models (i = j = k) \\
 \langle 0, 0, \star \rangle & \text{if } S \models (i + 1 = j, j + 1 = k) \\
 \langle 0, 0, \star \rangle & \text{if } S \models (i + 1 = k, k + 1 = j) \\
 \langle -2, 1, \star \rangle & \text{if } S \models (j = k \wedge j = i + 1).
 \end{cases}$$

Note that in this example we do not distinguish the global event and the global state since a global event always leads to a unique global state.

### III. VERIFICATION

In this section, we present our algorithm to verify whether a probabilistic BAR system modelled in the previous specification framework is a Nash-equilibrium. The input of the algorithm is  $m$  Byzantine players (possibly empty), denoted as  $\mathcal{M}^b = \{\mathcal{M}_1^b, \mathcal{M}_2^b, \dots, \mathcal{M}_m^b\}$ ,  $n - m$  non-Byzantine players (non-empty), either altruistic, denoted as  $\mathcal{M}^a = \{\mathcal{M}_{m+1}^a, \mathcal{M}_{m+2}^a, \dots, \mathcal{M}_n^a\}$ , or rational, denoted as  $\mathcal{M}^r = \{\mathcal{M}_{m+1}^r, \mathcal{M}_{m+2}^r, \dots, \mathcal{M}_n^r\}$ . We assign to each altruistic player  $\mathcal{M}_{m+j}^a \in \mathcal{M}^a$  a corresponding rational player  $\mathcal{M}_{m+j}^r \in \mathcal{M}^r$ .

We verify whether for any altruistic player, if she deviates from the specified behaviour, i.e., the altruistic player becomes a rational player, her pay-off would not increase. This property captures that the specified altruistic behaviour is a Nash-equilibrium, which is formally defined as follows.

**Nash-equilibrium** A specification  $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle$  with perfect information is a Nash-equilibrium if,

$$\forall \text{init} \in I, \forall i > m, U_i(\text{init}) \geq V_i(\text{init}). \quad (1)$$

A specification  $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle$  with imperfect information is a Nash-equilibrium if,

$$\forall i > m, \forall \text{init} \in I_i, U_i(\text{init}) \geq V_i(\text{init}). \quad (2)$$

$\mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r$  are defined the same as the input of the algorithm.  $I$  is the set of global initial states, i.e., all the combinations of the initial states of the total  $n$  players.  $m$  is the number of Byzantine players,  $i$  is the index of a player, since the Byzantine players are indexed from 1 to  $m$ ,  $i > m$  means that the player  $i$  is not a Byzantine player. In a perfect information game,  $U_i : S \times \mathbb{N} \rightarrow \mathbb{R}$  denotes the minimum guaranteed pay-off for player  $i$  starting from global state  $s \in S$  and doing exactly  $k \in \mathbb{N}$  actions, when  $i$  is *altruistic*.  $V_i : S \times \mathbb{N} \rightarrow \mathbb{R}$  denotes the maximum guaranteed expected pay-off achieved by player  $i$ , starting from global state  $s \in S$  and doing exactly  $k \in \mathbb{N}$  actions, when  $i$  is *rational*. When  $k$  tends to  $\infty$ , we simply denote the converged value as  $V_i(s)$  and  $U_i(s)$  respectively. Hence,  $U_i(\text{init}) \geq V_i(\text{init})$  ensures that a rational player does not deviate because the player would not gain pay-off by deviating. For imperfect information games, the minimum guaranteed expected pay-off of an altruistic player and maximum guaranteed pay-off of a rational player are defined locally and are denoted as  $U_i : S_i \times \mathbb{N} \rightarrow \mathbb{R}$  and  $V_i : S_i \times \mathbb{N} \rightarrow \mathbb{R}$ , respectively.

$U_i(\text{init})$  and  $V_i(\text{init})$  or  $U_i(\text{init})$  and  $V_i(\text{init})$  may not always be comparable. In order to be able to compare the two values, we restrict to games satisfying 1) finite states 2) pre-defined pay-offs 3) all strategies terminate with probability reaching 1.

Given a specification of a BAR system, i.e., a set of players,  $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle$ , the maximum non-terminating probability bound  $\epsilon$  with  $0 < \epsilon < 1$ , and a boolean value *perfect* indicating whether it is a perfect information game. The algorithm 1 first calculates the iteration steps *iter* by ensuring that the non-terminating probability is smaller than  $\epsilon$ . Then

we call the algorithm 2 by providing  $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle$  and *iter* as parameters. The algorithm 2 decides whether  $\langle \mathcal{M}^b, \mathcal{M}^a \rangle$  is a Nash-equilibrium. There are three possible outcomes of algorithm 2: *PASS* meaning that the specification is a Nash-equilibrium, *FAIL* meaning that the specification is NOT a Nash-equilibrium, and *TRY SMALLER  $\epsilon$*  meaning that the error is too large. In order to reduce the error,  $\epsilon$  needs to be set to a smaller value, so that more iteration steps are involved and thus the algorithm would terminate with a definite answer with a higher probability. (Details of algorithm 2 will be explained in the next paragraph). If the result is *TRY SMALLER  $\epsilon$* , we call the algorithm 1 with a smaller non-terminating bound  $\frac{\epsilon}{2}$ . To ensure that the loop terminates, we set a large iteration bound  $iter_{max}$ . When the iteration bound is reached without any definite answers, the algorithm 1 returns *NOT DECIDED* meaning that the algorithm cannot terminate with a definite result.

---

**Algorithm 1** *VerifyNash*( $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle$ ,  $\epsilon$ ,  $iter_{max}$ , *perfect*)

---

```

1: Let  $p_{min} = \min_{s \in S, a \in A} \{P(s, a)\}$ 
2:  $iter = \lceil \frac{\log(\epsilon)}{\log(1-p_{min})} \rceil$ 
3: if  $iter > iter_{max}$  then
4:   return NOT DECIDED
5: end if
6:  $Result \leftarrow CheckNash(\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle, iter, perfect)$ 
7: if  $Result = TRY\_SMALLER\ \epsilon$  then
8:    $VerifyNash(\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle, \frac{\epsilon}{2}, perfect)$ 
9: else
10:  return  $Result$ 
11: end if
```

---

The key part of algorithm 2 is to calculate the  $V_i(\text{init}, iter)$ ,  $U_i(\text{init}, iter)$  for perfect information game and calculate  $V_i(\text{init}, iter)$  and  $U_i(\text{init}, iter)$  for imperfect information game. Firstly, we initialize the four values to be 0 (line 9 and line 10). Then we iteratively calculate the four values. In each iteration, we update the expected pay-offs of the current iteration (step  $t$ ) to the corresponding values accumulated in the previous  $t - 1$  steps (line 11 to line 16). In detail, to calculate  $V_i$ , we first compute a set of possible expected pay-offs at a global state for each action combination of Byzantine players and rational player  $i$ . This set is then grouped by the possible next local states of rational player  $i$  ( $s_i \in S_i$ ). For each possible  $s_i$ , the Byzantine players together choose a combined action for the next state which minimises the expected pay-off of  $i$ . This gives the guaranteed pay-off at a given global state for each next local state  $s_i$  of player  $i$ . Then player  $i$  can choose the  $s_i$  which gives the maximum guaranteed pay-off at global state  $s$ . The difference in calculating  $U_i$  is that we calculate the expected pay-off over all the possible next local states for the player  $i$  at a global state, instead of taking the maximum, because an altruistic player does not make choice but just follows the specification. Compare to  $V_i(\text{init}, iter)$  and  $V_i(\text{init}, iter)$ , we

calculate  $Vl_i(\text{init}, \text{iter})$  and  $Ul_i(\text{init}, \text{iter})$  using a different reward function in the algorithm, which is the transition based reward structure ( $AH_i$ ). Given the local state  $ls$ ,  $Vl_i(ls, \text{iter})$  and  $Ul_i(ls, \text{iter})$  represent the expected pay-off for rational and altruistic, over all possible global states which contain  $ls$ .

---

**Algorithm 2** *CheckNash*( $\langle \mathcal{M}^b, \mathcal{M}^a, \mathcal{M}^r \rangle, \text{iter}, \text{perfect}$ )

---

```

1: for all  $i \in Al$  do
    #  $Al$  is the set of indexes of non-Byzantine players
2:   Let  $s \in S$ 
3:   Let  $ls \in S_i$ 
4:   Let  $ha_{max} = \max\{|AH_i^r(s, a)| | s \in S \wedge a \in A\}$ 
5:   Let  $hs_{max} = \max\{|SH_i^r(s)| | s \in S\}$ 
6:   Let  $p_{min} = \min_{s \in S, a \in A} \{P(s, a)\}$ 
7:    $max_{Aerror} = 2ha_{max}(1 - p_{min})^{\text{iter}}$ 
8:    $max_{Serror} = 2hs_{max}(1 - p_{min})^{\text{iter}}$ 
9:    $V_i(s, 0) \leftarrow 0; U_i(s, 0) \leftarrow 0;$ 
10:   $Vl_i(ls, 0) \leftarrow 0; Ul_i(ls, 0) \leftarrow 0;$ 
11:  for  $t=1$  to  $\text{iter}$  do
12:     $V_i(s, t) \leftarrow \max_{s'_i \in S_i} \{ \min_{s'_z \in S_z} E_{s'_i \setminus \{i\} \in S_{Al \setminus \{i\}}}$ 
       $\{SH_i^r(s') + V_i(s', t-1) | T(s, a) = s' \text{ for some } a \in A\} \}$  #  $Z$  is the set of indexes of Byzantine players
13:     $U_i(s, t) \leftarrow E_{s'_i \in S_i} \{ \min_{s'_z \in S_z} E_{s'_i \setminus \{i\} \in S_{Al \setminus \{i\}}}$ 
       $\{SH_i^a(s') + U_i(s', t-1) | T(s, a) = s' \text{ for some } a \in A\} \}$ 
14:     $Vl_i(ls, t) \leftarrow \max_{a_i \in A_i} \{ \{ E_{\{s \in S | s_i = ls\}} \min_{a_z \in A_z} \{$ 
       $E_{a_i \setminus \{i\} \in A_{Al \setminus \{i\}}} \{AH_i^r(s, a) + V_i(s', t-1) | T(s, a) =$ 
       $s' \wedge s_i = ls\} \} \}$ 
15:     $Ul_i(ls, t) \leftarrow E_{a_i \in A_i} \{ E_{\{s \in S | s_i = ls\}} \min_{a_z \in A_z} \{$ 
       $E_{a_i \setminus \{i\} \in A_{Al \setminus \{i\}}} \{AH_i^a(s, a) + U_i(s', t-1) | T(s, a) =$ 
       $s' \wedge s_i = ls\} \} \}$ 
16:  end for
17: end for
18: if !perfect then
19:   if  $\forall i \in Al, \forall \text{init} \in I_i. Vl_i(\text{init}, \text{iter}) \leq$ 
      $Ul_i(\text{init}, \text{iter}) \wedge Ul_i(\text{init}, \text{iter}) - Vl_i(\text{init}, \text{iter}) \geq$ 
      $max_{Aerror}$  then
20:    return PASS
21:   else if  $\exists i \in Al, \forall \text{init} \in I_i. Vl_i(\text{init}, \text{iter}) >$ 
      $Ul_i(\text{init}, \text{iter}) \wedge Vl_i(\text{init}, \text{iter}) - Ul_i(\text{init}, \text{iter}) >$ 
      $max_{Aerror}$  then
22:    return FAIL
23:   end if
24: end if
25: if perfect then
26:   if  $\forall i \in Al, \forall \text{init} \in I. V_i(\text{init}, \text{iter}) \leq U_i(\text{init}, \text{iter}) \wedge$ 
      $U_i(\text{init}, \text{iter}) - V_i(\text{init}, \text{iter}) \geq max_{Serror}$  then
27:    return PASS
28:   else if  $\exists i \in Al, \forall \text{init} \in I. V_i(\text{init}, \text{iter}) >$ 
      $U_i(\text{init}, \text{iter}) \wedge V_i(\text{init}, \text{iter}) - U_i(\text{init}, \text{iter}) >$ 
      $max_{Serror}$  then
29:    return FAIL
30:   end if
31: end if
32: return TRY SMALLER  $\epsilon$ 

```

---

Once the four values are calculated, we can use them to perform the verification. If the game is an imperfect information game, Nash-equilibrium definition 2 is applied. If it is a perfect information game, Nash-equilibrium definition 1 is applied. Since we limit the iteration steps, the above calculated value is an approximation of the pay-offs in infinite steps. That implies that an error exists. We calculate the maximum error bound in line 4 to line 8. Intuitively, we obtain the maximum absolute error of pay-off of the player  $i$ , i.e.,  $ha_{max}$  and  $hs_{max}$ , and then calculate the upper bound of the pay-off error in any infinite path starting from step  $\text{iter}$  (we denote the upper bound as  $e$ ), by summing up the pay-offs weighted by the probability of the corresponding actions in the path. The maximum error bound is computed by doubling the upper bound of pay-off error  $e$  due to the use of absolute value. The reason is that  $U_i(\text{init})$  is in the range of  $[U_i(\text{init}, \text{iter}) - e, U_i(\text{init}, \text{iter}) + e]$ ;  $V_i(\text{init})$  is in the range of  $[V_i(\text{init}, \text{iter}) - e, V_i(\text{init}, \text{iter}) + e]$ ; by ensuring  $U_i(\text{init}, \text{iter}) - e \geq V_i(\text{init}, \text{iter}) + e$ , we ensure that  $U_i(\text{init}) \geq V_i(\text{init})$ . Given the calculated error bound, we compare  $|V_i(\text{init}, \text{iter}) - U_i(\text{init}, \text{iter})|$  with the error bound (or compare  $|Vl_i(\text{init}, \text{iter}) - Ul_i(\text{init}, \text{iter})|$  with the error bound). If  $|V_i(\text{init}, \text{iter}) - U_i(\text{init}, \text{iter})|$  (or  $|Vl_i(\text{init}, \text{iter}) - Ul_i(\text{init}, \text{iter})|$ ) is no less than the error bound, meaning that the error is NOT too large, the algorithm terminates; otherwise the algorithm returns TRY SMALLER  $\epsilon$ . When the algorithm terminates, if  $V_i(\text{init}) \geq U_i(\text{init})$ , the algorithm returns PASS, meaning that the specification is a Nash-equilibrium, otherwise, the algorithm returns FAIL meaning that the specification is not a Nash-equilibrium.

Theoretically, if the iteration number is not bounded with  $\text{iter}_{max}$ , the algorithm should terminate with PASS or FAIL, for any scenario where  $V_i(\text{init}) - U_i(\text{init})$  converges to a non-zero value. If  $V_i(\text{init}) - U_i(\text{init})$  converges to zero, then the algorithm is not guaranteed to terminate because, it terminates only if  $|V_i(\text{init}, \text{iter}) - U_i(\text{init}, \text{iter})| = max_{Serror}$  ( $|Vl_i(\text{init}, \text{iter}) - Ul_i(\text{init}, \text{iter})| = max_{Aerror}$  correspondingly) for some  $\text{iter}$  value.

#### IV. EXPLANATION OF THE ALGORITHMS

##### A. Algorithm 1

In line 1 the algorithm calculates the minimum value of the probabilities for all global transitions. The probability for one global transition is the product of each local probability. Since we calculate the minimum probability, we consider the case that all the non-Byzantine players are altruistic.

We estimated the sufficient number iterations ( $\text{iter}$ ) for verification as follows (in line 2). Before that  $\text{iter}$  number of steps the termination probability has an upper bound which should be smaller than  $\epsilon$  denoted as,  $(1 - p_{min})^{\text{iter}} < \epsilon$ . From the above formula  $\text{iter}$  is derived as,  $\text{iter} = \lceil \frac{\log(\epsilon)}{\log(1-p_{min})} \rceil$ . From line 3 to line 5 if  $\text{iter} > \text{iter}_{max}$  it means that the algorithm cannot make a decision within the maximum allowed iteration steps. Therefore it returns NOT DECIDED. If  $\text{iter} \leq \text{iter}_{max}$  we call algorithm 2 and store the returned value in Result (line 6). If the value is TRY SMALLER  $\epsilon$  we call the algorithm 1 by changing  $\epsilon$  to  $\frac{\epsilon}{2}$  (line 7 to 8). Otherwise,

the returned value is a definite answer on whether the system is a Nash-equilibrium. Therefore, we report the returned value (line 10).

### B. Algorithm 2

For every non-Byzantine player  $i$ , we first calculate maximum absolute transition pay-off (line 4), maximum absolute state pay-off (line 5), minimum value of probability of global transition (line 6). Note that we use  $AH_i^r$  and  $SH_i^r$  in line 4 and 5 respectively, because,  $AH_i^r$  contains  $AH_i^a$  ( $SH_i^r$  contains  $SH_i^a$ ) due to fact that rational player has more states and transitions.

In line 7 and 8, we calculate the upper bound of the size of range of uncertainty of the expected transition pay-off and state pay-off in  $iter$  steps respectively. Then we initialize  $V_i$ ,  $U_i$ ,  $Vl_i$  and  $Ul_i$  to 0 (line 9 and 10).

From line 12 to 16 we update the four values iteratively by using a dynamic programming algorithm. For any global state  $s$ ,  $t^{th}$  iteration in the algorithm updates the optimized pay-offs that can be obtained in  $t$  steps starting from the state  $s$ . For perfect information games, we do a pre-process to transfer each transition pay-off to the state pay-off that the transition leads to. Similarly, for imperfect information, we transfer each state pay-off to the transition pay-off that the transition leads to the state.

In line 12, for state  $s$  in the  $t^{th}$  iteration we first 1) add the pay-off from immediate transitions from  $s$  to  $s'$  to the corresponding optimal pay-offs calculated at state  $s'$  for  $t-1$  steps (this value has been calculated at the previous iteration) and the result is denoted by  $v$ . Since, there can be many  $s'$ s that  $s$  can transit to we actual have a set of values. We 2) group a set of  $vs$  by fixing the choice of Byzantine and rational players and calculate the expected value ( $ev$ ) of this set denoted as  $E_{s'_{Al} \setminus \{i\} \in S_{Al} \setminus \{i\}}$ . Now we have a set of expected values each corresponding to a choice of Byzantine and rational players. We again 3) group a set of  $ev$ 's by fixing the rational choice and calculate the minimum of the set and call it  $bev$ . This means for any choice by the rational player, Byzantine player chooses his action such that the rational player's pay-off is minimized. Finally, we 4) find the maximum value of the set of  $bevs$  and assign it to  $V_i(s, t)$ . This means rational player chooses his action such that his pay-off is maximised in this restricted set.

In line 13, we calculate  $U_i(s, t)$  which is the optimal expected pay-off of the altruistic player  $i$  from the state  $s$  after  $t$  steps. The calculation is similar to the previous line. The only difference is that, in the final grouping step we calculate the expected value instead of the maximum value because, we are considering altruistic players which do not choose between actions but follow the probabilistic distribution.

Line 14 is similar to line 12. The differences are we use  $AH_i^r$  instead of  $SH_i^r$  and we add a step between 2) and 3) and denote the new step as 2'). In step 2'), since we are considering imperfect information games there is a set of equivalent global states for a given local state of player  $i$  we have multiple sets of  $bevs$ . We need to calculate the expected

value (denoted as  $\{E_{\{s \in S | s_i = l_s\}}\}$ ) for each set as the  $bev$  for the final step.

Similarly, in line 15 we add the additional step to line 13 and calculate  $Ul_i(s, t)$  using  $AH_i^r$  instead of  $SH_i^r$ .

After updating the four values in  $iter$  steps we can decide whether the system is Nash-Equilibrium. If the game is imperfect information, after  $iter$  iterations for all altruistic players, if the  $Vl_i(init, iter)$  – optimal rational pay-off starting from the local initial state  $init$  and playing for  $iter$  steps – is not larger than  $Ul_i(init, iter)$  – the altruistic pay-off – and the difference between  $Ul_i(init, iter)$  and  $Vl_i(init, iter)$  is not smaller than  $max_{Aerror}$  then the system is a Nash-equilibrium, thus the system returns *PASS* (line 19 and 20). The conditions ensures that all the possible values for  $Vl_i(init)$  are not larger than any possible value of  $Ul_i(init)$ . Therefore, any non-Byzantine player does not deviate from the protocol. Similarly, the conditions ensure that –add conditions–  $Vl_i(init)$  is larger than  $Ul_i(init)$  for some  $i$ . This means that, a player may deviate from the protocol specification and thus the protocol is not a Nash-equilibrium (line 20-22). The reasoning is similar for perfect information games (line 25-29). If none of the above conditions are satisfied means that the error bound is too large i.e. the  $iter$  is too small. In this case we return *TRY SMALLER  $\epsilon$* . A smaller  $\epsilon$  would lead to more iteration steps and lower  $max_{Aerror}$  and  $max_{Serror}$  which increases the probability of finding a definite answer.

## V. CASE STUDY

We apply our algorithm on a case study with perfect information, the three-player Rock-Paper-Scissors, and on another with imperfect information, the secret sharing protocol from Halpern and al. [7].

### A. Rock-Paper-Scissors game

We analyse two strategies of the Rock-Paper-Scissors game in Example 1: each player has equally distributed probability on playing rock, paper and scissors (1/3 for playing rock, 1/3 for playing paper, and 1/3 for playing scissors); and each player has unequal probability distribution (1/5 for playing rock, 1/5 for playing paper, and 3/5 for playing scissors). For each strategy, we analyse two scenarios: no Byzantine players, and having one Byzantine player.

#### 1) Equal probability strategy:

a) *No Byzantine Players:* When there is no Byzantine players ( $m = 0$ ), to verify whether the equally distributed probability strategy is a Nash-equilibrium, we actually need to verify the following conditions, according to the definition of Nash-equilibrium in the section III.

- 1)  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle) \geq V_1(\langle s_1^r, s_1^a, s_1^a \rangle)$
- 2)  $U_2(\langle s_1^a, s_1^a, s_1^a \rangle) \geq V_2(\langle s_1^a, s_1^r, s_1^a \rangle)$
- 3)  $U_3(\langle s_1^a, s_1^a, s_1^a \rangle) \geq V_3(\langle s_1^a, s_1^a, s_1^r \rangle)$

Symbol  $s_1^a$  is the initial state for an altruistic player as defined in the altruistic player modelling in Section II-A. Similarly,  $s_1^r$  is the initial state for a rational player. Assuming three altruistic players, the global initial state is  $\langle s_1^a, s_1^a, s_1^a \rangle$ . When the first

player is a rational player, the initial global state is  $\langle s_1^r, s_1^a, s_1^a \rangle$ .  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle)$  is the maximum guaranteed expected pay-off for player  $p_1$  when  $p_1$  is altruistic;  $V_1(\langle s_1^r, s_1^a, s_1^a \rangle)$  is the maximum guaranteed expected pay-off for player  $p_1$  when  $p_1$  is rational, i.e., may deviate from the equal probability strategy. Hence, the first condition says that player  $p_1$  would not deviate from the specified strategy since the deviation pay-off is smaller. Similarly, the second condition ensures that  $p_2$  would not deviate and the third condition ensures that  $p_3$  would not deviate.

Since the behaviour of all altruistic players are exactly the same, the three conditions are symmetrical. Hence, it is sufficient to prove one of the above conditions. Assume the game ends, if and only if there is a unique winner. The verification result shows that the equal probabilistic strategy is a Nash-equilibrium. We provide the intuition of the result by illustrating the first condition, i.e., reasoning on player  $p_1$ 's behaviour, as follows.

For each iteration, we need to consider the following two cases, corresponding to  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle)$  and  $V_1(\langle s_1^r, s_1^a, s_1^a \rangle)$  respectively. We show that the probability of ending the game in both of the two cases are the same and the pay-off of  $p_1$  in both cases are the same as well.

- *Player  $p_1$  is altruistic.* Since each player has three actions (plying rock, paper, or scissors) and the probability of each action is  $\frac{1}{3}$ , there are in total 27 outcomes and each outcome has probability  $\frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$ . Among all the outcomes, there are 9 outcomes that lead to a unique winner, which end the game. Thus, the probability that the game ends in one iteration is  $9 \times \frac{1}{27} = \frac{1}{3}$ . The expected pay-off of  $p_1$  is  $\frac{1}{27} \times ep_a^1 + \dots + \frac{1}{27} \times ep_a^{27}$  where  $ep_a^i$  is the expected pay-off for outcome  $i$ . We show that the value of the expected pay-off is 0. For instance, if we fix  $p_1$ 's action as playing rock 0, the outcomes together with the pay-offs are showing in Table I. Since, the probability of each outcome is the same, the expected pay-off in this case is obviously 0 (Pay-offs cancel out, see Table I). Similarly when  $p_1$  plays other actions. Hence the total pay-off for  $p_1$  is 0.
- *Player  $p_1$  is rational.* Since  $p_1$  is rational, he can choose any action with any probability. To show the ending probability of the game, we show that for any action of  $p_1$ , the probability to end the game is exactly  $\frac{1}{3}$  as follows. Given a certain action of  $p_1$ , there are 9 outcomes and each outcome has probability  $\frac{1}{3} \times \frac{1}{3} = \frac{1}{9}$ . Among these 9 outcomes, there are 3 outcomes with a unique winner. Hence, the probability of having a unique winner for any choice of player  $p_1$  is  $3 \times \frac{1}{9} = \frac{1}{3}$ . Similarly, for calculating the pay-offs, given one action of  $p_1$ , her expected pay-off is 0 (see from Table I). Hence, for any probabilistic distribution of  $p_1$ 's actions, the expected pay-off is 0.

When the iteration is infinite,  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle)$  is calculated as  $ep_a + \frac{2}{3} \times ep_a + \frac{2^2}{3} \times ep_a + \dots$  where  $ep_a$  is the expected pay-off of each iteration,  $\frac{2}{3}$  is the probability of continue the game. Since  $ep_a$  is 0,  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle) = 0$ .

TABLE I  
PAY-OFFS OF PLAYER  $p_1$  WHEN PLAYING ROCK

|         |   |    |   |    |    |   |   |   |   |
|---------|---|----|---|----|----|---|---|---|---|
| i       | 0 | 0  | 0 | 0  | 0  | 0 | 0 | 0 | 0 |
| j       | 0 | 0  | 0 | 1  | 1  | 1 | 2 | 2 | 2 |
| k       | 0 | 1  | 2 | 0  | 1  | 2 | 0 | 1 | 2 |
| pay-off | 0 | -1 | 1 | -1 | -2 | 0 | 1 | 0 | 2 |

TABLE II  
PAY-OFFS OF PLAYER  $p_2$  WHEN  $p_1$  PLAYS ROCK

|         |   |    |   |   |   |   |    |   |    |
|---------|---|----|---|---|---|---|----|---|----|
| i       | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0  |
| j       | 0 | 0  | 0 | 1 | 1 | 1 | 2  | 2 | 2  |
| k       | 0 | 1  | 2 | 0 | 1 | 2 | 0  | 1 | 2  |
| pay-off | 0 | -1 | 1 | 2 | 1 | 0 | -2 | 0 | -1 |

Similarly,  $V_1(\langle s_1^r, s_1^a, s_1^a \rangle) = 0$ . Thus, the first condition  $U_1(\langle s_1^a, s_1^a, s_1^a \rangle) \geq V_1(\langle s_1^r, s_1^a, s_1^a \rangle)$  is satisfied. The other two conditions can be proved in a similar way, especially they are symmetrical to the first condition.

b) *One Byzantine Player:* Assume player  $p_1$  is Byzantine ( $m = 1$ ), to verify whether the equally distributed probability strategy is a Nash-equilibrium, we need to verify the following conditions, according to the definition of NE in section III.

- 1)  $U_2(\langle s_1^b, s_1^a, s_1^a \rangle) \geq V_2(\langle s_1^b, s_1^r, s_1^a \rangle)$
- 2)  $U_3(\langle s_1^b, s_1^a, s_1^a \rangle) \geq V_3(\langle s_1^b, s_1^r, s_1^a \rangle)$

The verification result shows that the equal probabilistic strategy is a Nash-equilibrium as well when  $p_1$  is Byzantine. The intuitions are as follows.

Similarly, for each iteration, we consider the following two cases. Like the previous case with no Byzantine player, the probability of ending the game in both cases are the same  $\frac{1}{3}$ . Unlike the previous case, the pay-offs in the two cases differ from each other.

- *Player  $p_2$  is altruistic.* Recall that the Byzantine player only has non-deterministic actions at each state. According to the algorithm, we always consider the action of the Byzantine player that leads to the worst case of pay-off for player  $p_2$ , i.e., we only consider the  $p_1$ 's action that minimize  $p_2$ 's expected pay-off. For each action of Byzantine player  $p_1$ , we calculate the expected pay-off of  $p_2$ . For example, if  $p_1$  plays rock 0, the pay-offs of  $p_2$  for each outcome are shown in Table II. The probability of each outcome is exactly  $\frac{1}{9}$ . Hence the expected pay-off of  $p_2$  is 0. Similarly, for the other two actions of  $p_1$ , the expected pay-offs of  $p_2$  are 0 as well.
- *Player  $p_2$  is rational.* Since  $p_2$  is rational, her actions can have any probabilistic distribution. Similar to the case with no Byzantine player, we consider the pay-off for each action of  $p_2$ . Technically, we consider the pay-off of  $p_2$  given any combination of  $p_1$ 's action and  $p_2$ 's action. Recall that we only need to consider  $p_1$ 's action that leads to the worse case for  $p_2$ . Given  $p_2$ 's action,  $p_1$ 's action is decided. For example, when  $p_2$  plays rock, the action of  $p_1$  that leads to the worst case for  $p_2$  is playing paper. The pay-offs when  $p_2$  plays rock for each outcome is given in Table III. Hence, the expected pay-off of  $p_2$  when playing



TABLE III  
PAY-OFFS OF PLAYER  $p_2$  WHEN PLAYING ROCK

|         |    |    |   |
|---------|----|----|---|
| i       | 1  | 1  | 1 |
| j       | 0  | 0  | 0 |
| k       | 0  | 1  | 2 |
| pay-off | -1 | -2 | 0 |

rock  $\frac{1}{3} \times (-1 + -2 + 0) = -1$ . For the other two actions of  $p_2$ , the expected pay-off is calculated in a similar way. Due to the symmetric property, the expected pay-off of  $p_2$  when playing the other two actions are  $-1$  as well. Hence, the expected pay-off of  $p_2$  is  $-1$ . Note that unlike the previous case with no Byzantine players, the rational player's pay-off decreased when a Byzantine player is presented.

Therefore,  $U_2(\langle s_1^b, s_1^a, s_1^a \rangle) = ep_a + \frac{2}{3} \times ep_a + \frac{2}{3} \times ep_a + \dots$ , where  $ep_a$  is the expected pay-off of  $p_2$  in each iteration. Since  $ep_a = 0$ , we have  $U_2(\langle s_1^b, s_1^a, s_1^a \rangle) = 0$ . Similarly,  $V_2(\langle s_1^b, s_1^r, s_1^a \rangle) = -1 + \frac{2}{3} \times (-1) + \frac{2}{3} \times (-1) + \dots \approx -3$  meaning that the value converges to  $-3$ . Thus, we have  $U_2(\langle s_1^b, s_1^a, s_1^a \rangle) \geq V_2(\langle s_1^b, s_1^r, s_1^a \rangle)$ . Similarly, we have  $U_3(\langle s_1^b, s_1^a, s_1^a \rangle) \geq V_3(\langle s_1^b, s_1^r, s_1^a \rangle)$ . It means that the equal probability strategy is a Nash-equilibrium, when  $p_1$  is Byzantine.

2) *Unequal probability strategy*: Given the probabilistic strategy of the altruistic player  $\langle \frac{1}{5}, \frac{1}{5}, \frac{3}{5} \rangle$ , we analyse the above two scenarios – without Byzantine players and with one Byzantine player. Differing from the case of equal probability strategy, in the unequal probability strategy, the probability of each outcome becomes unequal as well. This introduces difficulties in calculation of the expected pay-offs. For the simplicity of calculation, we group the outcomes with the same probability and obtain the following classes, such that we do not need to consider the total 27 outcomes every time.

- Unique winner and the winner plays 0
- Unique winner and the winner plays 1
- Unique winner and the winner plays 2
- Unique loser and the loser plays 0
- Unique loser and the loser plays 1
- Unique loser and the loser plays 2
- Each player plays 0
- Each player plays 1
- Each player plays 2
- other cases

a) *Without Byzantine*: Similarly, we show that in each of the above classes, the expected pay-off of any altruistic player is 0. For example, in the first class – unique winner and the winner plays 0, there are 3 cases:  $p_1$  wins,  $p_2$  wins, or  $p_3$  wins. An altruistic player  $p_1$ 's pay-offs are 2,  $-1$  and  $-1$  respectively. In this class the probability of each outcome is the same. Thus, the expected pay-off of  $p_1$  is 0. Similarly, other players' pay-offs are 0 in the first class as well. The expected pay-offs of any altruistic player is 0 in other classes. Therefore, considering all the classes, the expected pay-off of any altruistic player is 0.

Next, we calculate the expected pay-off of a rational player using a similar way as in the equal probability strategy. Assuming  $p_1$  is rational, given that  $p_1$  plays rock, there are 9 outcomes. The pay-off of each outcome is shown in Table I. However, the probability of each outcome is unequal. We calculate the probabilities for each outcome and obtain the expected pay-off of  $p_1$  is 0.8, and calculate the probability of restarting the game as  $\frac{86}{125}$ . Hence, for each iteration, the expected pay-off of  $p_1$  is no less than 0.8. If any other action of  $p_1$  leads to a pay-off less than 0.8,  $p_1$  will play rock instead of the action for achieving better pay-off. Hence, the expected pay-off of  $p_1$  in each iteration is no less than 0.8. For infinite number of iterations, the expected pay-off of  $p_1$  is obviously larger than 0. This implies that  $V_1(\langle s_1^r, s_1^a, s_1^a \rangle) > U_1(\langle s_1^a, s_1^a, s_1^a \rangle)$ . Therefore, the strategy  $\langle \frac{1}{5}, \frac{1}{5}, \frac{3}{5} \rangle$  is NOT a Nash-equilibrium, when there is no Byzantine player.

b) *With one Byzantine player*: Assume player  $p_1$  is Byzantine, unlike the case with equal probability strategy, the unequal probability strategy provides interesting and counter-intuitive scenarios. We cannot simply consider one iteration and derive the Nash-equilibrium by extending it to multiple iterations. Essentially, the probability of ending the game plays more important role in the calculation. An action of rational player which leads to bigger pay-off than other actions in one iteration may not always lead to a bigger pay-off in the long run, in the case that the action leads to a significant larger probability to restart the game. For example, in the case that the pay-off is negative, if player's pay-off in the long run is smaller than the pay-off in one iteration then the action that has smaller pay-off in one iteration has larger pay-off in the long run.

In the algorithm,  $U_i(\langle s_1^b, s_1^a, s_1^a \rangle, iter)$  was approximately equal to  $\frac{-10}{11}$  when  $p_2$  is altruistic and  $V_i(\langle s_1^b, s_1^r, s_1^a \rangle, iter)$  was approximately equal to  $\frac{-5}{3}$  when  $p_2$  was rational. The algorithm terminated by returning the result *PASS*. It shows that the strategy  $\langle \frac{1}{5}, \frac{1}{5}, \frac{3}{5} \rangle$  is a Nash-equilibrium, when  $p_1$  is Byzantine. Due to space limitations we do not present the proofs in the paper. Please refer to [8] for proofs.

## B. Probabilistic secret sharing

Secret sharing is a way for a group of users to share a secret: the secret is divided into many parts, named *shares*, and each user in the group has a share; the secret can only be reconstructed with a sufficient number (threshold  $m$ ) of shares. In other words, the secret is kept confidential, even in the presence of a limited number of traitors (compromised users or non-cooperating members who prevent the reconstruction of the secret). This property makes secret sharing an ideal scheme for storing highly sensitive information, such as encryption keys and launch codes. In the parametric form, a  $n$ - $m$  ( $n \geq m$ ) secret sharing ensures that it requires at least  $m$  altruistic players to reconstruct the secret within a group of  $n$  players. That is, less than  $m$  compromised players together cannot reconstruct the secret; and less than  $n - m$  non-cooperating members together cannot stop the altruistic players from reconstructing the secret.

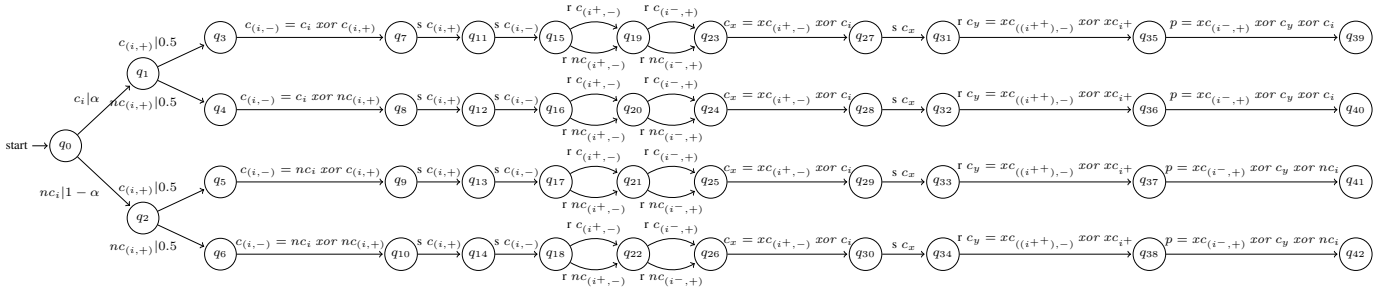


Fig. 1. Altruistic player in probabilistic secret sharing

The most well-known secret sharing scheme is the Shamir's secret sharing [9]. However, the scheme does not work when the players are rational. It is shown that a rational player will not share her secret with anyone else [10]. In fact, it is proved that in general, in all practical mechanisms for shared-secret reconstruction with an upper bound on the running time, the best strategy for each rational player is to deviate from the specification by doing nothing [10]. Hence, a randomized mechanism (the probabilistic secret sharing scheme) is proposed to ensure that rational players would reveal their shares to reconstruct the secret, i.e., achieving a Nash-equilibrium with a constant expected running time.

1) *Protocol description:* The probabilistic scheme is a 3-3 secret sharing. Given the 3 players, we index each player with a distinct number in  $\{1, 2, 3\}$ . We use  $i$  to denote one player, then the other two players are  $(i+1)\%3$ , denoted as  $i^+$ , and  $(i-1)\%3$ , denoted as  $i^-$ . There is an issuer who assigns each player a share of secret. Assuming the share cannot be changed; for example, it is signed by the issuer. The scheme works as follows.

1) Each player  $i$

- chooses a bit  $c_i$  ( $c_i$  can only be 1 or 0) such that  $c_i = 1$  with probability  $\alpha$  and  $c_i = 0$  with probability  $1 - \alpha$ ;
- chooses a bit  $c_{(i,+)}$  ( $c_{(i,+)}$  can only be 1 or 0) randomly, so that  $c_{(i,+)} = 1$  with probability  $1/2$  and  $c_{(i,+)} = 0$  with probability  $1/2$ ;
- calculates  $c_{(i,-)} = c_i \text{ xor } c_{(i,+)}$ .

Then player  $i$  sends the bit  $c_{(i,+)}$  to player  $i^+$  and sends the bit  $c_{(i,-)}$  to player  $i^-$ . This means that the player  $i$  receives a bit  $c_{(i^+,-)}$  from player  $i^+$  and a bit  $c_{(i^-,+)}$  from player  $i^-$ .

- 2) On receiving  $c_{(i^+,-)}$ , player  $i$  computes  $c_{(i^+,-)} \text{ xor } c_i$  and sends it to player  $i^-$ . Correspondingly,  $i$  receives  $c_{((i^+)+,-)} \text{ xor } c_{i^+}$  from  $i^+$ . Since there are only 3 players, player  $(i^+)^+$  is the player  $i^-$ . Hence  $c_{((i^+)+,-)} \text{ xor } c_{i^+} = c_{(i^-,+)} \text{ xor } c_{i^+}$ .
- 3) When receiving both  $c_{(i^-,+)}$  from player  $i^-$  (step 1) and receiving  $c_{(i^-,+)} \text{ xor } c_{i^+}$  from  $i^+$  (step 2),  $i$  computes  $p = c_{(i^-,+)} \text{ xor } (c_{(i^-,+)} \text{ xor } c_{i^+}) \text{ xor } c_i$ . Note that  $c_{(i^-,+)} \text{ xor } (c_{(i^-,+)} \text{ xor } c_{i^+}) \text{ xor } c_i = c_{i^-} \text{ xor } c_{i^+} \text{ xor } c_i$ , i.e.,  $p = c_1 \text{ xor } c_2 \text{ xor } c_3$  (It holds for all players). If  $p = c_i = 1$  then player  $i$  sends her share to the others.
- 4) Depending on the value of  $p$  and the shares received,

player  $i$  constructs the secret, restarts the protocol or reports cheating.

- If  $p = 1$ ,  $i$  receives 3 shares, then he construct the secret.
- If  $p = 0$  and  $i$  received no secret shares, or if  $p = 1$  and  $i$  received exactly one share (possibly from itself; that is, we allow the case that  $i$  did not receive any shares from other players but sent its own), the issuer restarts the protocol.
- Otherwise, player  $i$  stops the protocol. Because someone must have been cheating.

The intuition behind the last step is as follows: When  $p = 0$ , no one should send her share; and thus  $i$  receives no secret share. Otherwise, someone calculated wrong (cheating). When  $p = 1$  either  $c_i = c_{i^+} = c_{i^-} = 1$  or there exists exactly one player chooses 1, i.e.,  $c_i = c_{i^+} = 0$  and  $c_{i^-} = 1$ ,  $c_i = c_{i^-} = 0$  and  $c_{i^+} = 1$ , or  $c_{i^+} = c_{i^-} = 0$  and  $c_i = 1$ . In the former case, each player can construct the secret. In the later case,  $i$  should receive exactly one share then the issuer restarts the protocol. Otherwise, some player calculated wrong (cheating).

2) *Modelling:* The model of the altruistic players who follow the protocol specification is straightforward. For the sake of space saving, we present the FMS for altruistic players in the graphical manner as in Figure 1. In the Figure, we use 's' to stand for sending and 'r' to stand for receiving, use  $nc_i$  to represent the opposite value of  $c_i$ ; the value after symbol  $|$  is the probability of the action, for example,  $c_i|\alpha$  stands for that the probability of choosing the value  $c_i$  is  $\alpha$ . For simplicity of presentation, we merged the states and traces caused by received values and use the prefix  $xc$  to represent either  $c$  or  $nc$  depending on which one is received. The payoff function is defined as follows:

$$SH_i^a(S) = \begin{cases} 3 & \text{if only the player } i \text{ gets secret} \\ 2 & \text{if the player } i \text{ and another player get secret} \\ 1 & \text{if everyone knows the secret} \\ 0 & \text{if no one knows the secret} \\ -1 & \text{if other two players know the secret} \\ -2 & \text{if only one other player knows the secret} \end{cases}$$

The Byzantine players deviate from the protocol specification by sending the opposite value of the one that the player should send. For example, in a Byzantine player's model, we add a transition from  $q_7$  to  $q'_{11}$  labelled with  $nc_{(i,+)}$ , and

copy the transition from  $q_{11}$  to  $q_{39}$  as  $q'_{11}$  to  $q'_{39}$ . Similar for other states which enables sending actions. Due to space limit, we omit the figure for Byzantine players. The model of rational players are the model of Byzantine players together with the above pay-off function.

Notice that we fixed the order of all players' sending and receiving actions such that the three players' sending and receiving actions can be synchronised. That is, we did not model the full interleaving of all the possible orders of sending and receiving actions of the three players. We show that this does not influence the verification result.

3) *Verification results:* We verified whether a player deviates from the specification in two settings: without Byzantine player and with one Byzantine player. Essentially, our algorithm calculates the pay-off of a player in the following four configurations and compare the pay-offs.

- 1) All the three players are altruistic ( $m = 0$ )
- 2) Two players are altruistic and one player is rational ( $m = 0$ )
- 3) Two players are altruistic and one player is Byzantine ( $m = 1$ )
- 4) One player is altruistic, one player is Byzantine and one player is rational ( $m = 1$ )

The verification result shows that the probabilistic secret sharing protocol is a Nash-equilibrium when there is no Byzantine player, which confirms the manual analysis in its original paper. In addition, the verification result shows that even with a Byzantine player, the probabilistic secret sharing scheme is still a Nash-equilibrium, which has not been proved in existing works.

## VI. EXAMPLES

It appears that the optimal move for a single iteration optimizes the expected pay-off in the long run. We have proved that for equal probability strategy, optimal moves for rational player and Byzantine player was the repetitive use of optimal move for each single iteration. For unequal probability strategy without Byzantine player we do not have to consider the optimal long-run strategy of a rational player to prove the Nash-equilibrium. (The reason is that optimal long-run expected pay-off for an altruistic player is 0 irrespective of the strategy, and the rational player has a strategy which results in a positive pay-off for each iteration. It is logical that optimal strategy of the rational player also results in a positive expected pay-off. In order to arrive at this conclusion we do not explicitly generate the optimal rational strategy.) When it comes to unequal probability strategy with Byzantine players, we need to calculate optimal strategies for rational players as well as altruistic players because, we do not have any logical inferencing to argue about the optimal long-run expected pay-offs without explicitly generating them. Now, the claim about repeating the move which optimized a single iteration to optimize the expected pay-off in the long-run, has become counter intuitive. We start by investigating the case where, we have unequal probability strategy with Byzantine player for a game having two iterations.

The analysis considers the case when  $p_1$  is Byzantine,  $p_2$  is rational and  $p_3$  is altruistic.  $p_1, p_2$  and  $p_3$  are defined in section V. We define following 6 symbols.  $\gamma_0, \gamma_1$  and  $\gamma_2$  are the restarting probabilities when  $p_2$  plays rock, paper, scissor moves respectively.  $X_0, X_1$  and  $X_2$  are the expected pay-offs for  $p_2$ , for rock, paper, scissor moves respectively.  $\gamma_0 = \frac{4}{5}, \gamma_1 = \frac{4}{5}$  and  $\gamma_2 = \frac{2}{5}$ .

We show the calculation for  $X_0$  here. Table III shows the pay-offs for a rational player in a single iteration when he plays rock. According to the table, expected pay-off for the first iteration is calculated as,

$\frac{1}{5} \times (-1) + \frac{1}{5} \times (-2) + \frac{3}{5} \times (-0) = \frac{-3}{5}$ . Hence,  $X_0 = \frac{-3}{5}$ .  $X_1$  and  $X_2$  are calculated in a similar way. The results are  $X_1 = \frac{-7}{5}$  and  $X_2 = -1$  respectively.

Consider two iterations, the expected pay-off for rational player  $p_2$  in two iterations is calculated as,

$X_i + P_i \times X_j$  where  $i, j \in \{0, 1, 2\}$ . Since  $\gamma_0 = \gamma_1$  and  $X_0 > X_1$ , rock move is more optimal than paper move in both iterations. Hence, we consider only the rock move and the scissor move for the analysis. Now we have four strategies to compare. (two moves for the first iteration  $\times$  two moves for the second iteration) For any given first move, the second move to optimize the summation would be the maximum between  $X_0$  and  $X_2$ . Now we have only two strategies to compare.

- $X_0 + \gamma_0 \times X_0 = \frac{-3}{5} + \frac{4}{5} \times \frac{-3}{5} = -1.44$

- $X_2 + \gamma_1 \times X_0 = -1 + \frac{2}{5} \times \frac{-3}{5} = -1.24$

We can see that second strategy becomes optimal as instead of repetitive rock strategy. This counter intuitive result is caused by the restarting probability. In case of negative pay-offs, even though a move has high expected pay-off values in a single iteration, higher restarting probabilities make them less profitable in the long-run and vice-versa.

We can deduce that the long-run optimal altruistic and rational strategies for an infinite number of iterations is not trivial. We show how to determine the optimal strategies in the long-run in our proof of Nash-equilibrium for the unequal strategy in Section VII.

## VII. THEOREMS AND PROOFS

**Theorem VII.1.** *Let  $\mathcal{M}$  be a mechanism defined in section VI. Let  $P_i^a(s_1^a, 0) = 1/5, P_i^a(s_1^a, 1) = 1/5, P_i^a(s_1^a, 2) = 3/5$ . Suppose there is one Byzantine player.  $P_i^a$  is a Nash-equilibrium*

First we have to show that, rational players strategy is always playing the scissor move.

We use the notation defined in section VI for the rest of the proof. To calculate the expected pay-off for the rational player in the long-run, we have to define restarting probabilities and expected pay-offs w.r.t. each iteration. For the  $i^{th}$  iteration there is a corresponding  $X^{(i)}$  term and a  $P^{(i)}$  term.  $X^{(i)}$  is the expected pay-off at  $i^{th}$  iteration.  $X^{(i)}$  can take any value among  $\{X_0, X_1, X_2\}$ .

$P^{(i)}$  is the restarting probability at  $i^{th}$  iteration.  $P^{(i)}$  can take

any value among  $\{\gamma_0, \gamma_1, \gamma_2\}$ . Since the restarting probability depends on the current move,  $X^{(i)}$  and  $P^{(i)}$  can be related.  $\forall k \in \{0, 1, 2\}, X^{(i)} = X_k \iff P^{(i)} = P_k$  Now we can define the weighted expected pay-off of the iteration  $i$ . Simply it should be the product of expected pay-off of the iteration  $i$  and the probability of reaching iteration  $i$ . Expected pay-off of the iteration  $i$  is  $X^{(i)}$ . probability of reaching iteration  $i$  is

$$Pr(i) = \begin{cases} 1 & \text{if } i = 1 \\ \prod_{k=1}^{i-1} P^{(k)} & \text{if } i > 1 \end{cases}$$

The long run pay-off is the infinite summation of all the weighted expected pay-offs is,

$$\sum_{i=1}^{\infty} X^{(i)} Pr(i).$$

**Lemma VII.2.** *The series  $\sum_{i=1}^{\infty} X^{(i)} Pr(i)$  Converges.*

*Proof.* We can prove the absolute convergence of the series which is sufficient to prove the convergence of the original series.

$$\sum_{i=1}^{\infty} |X^{(i)} Pr(i)| \\ \iff \sum_{i=1}^{\infty} |X^{(i)}| |Pr(i)|$$

Let's consider the summation of  $n$  terms in the series  $(\sum_{i=1}^n |X^{(i)}| |Pr(i)|)$  as the  $n^{th}$  term in a sequence. Then we have to prove the convergence of this sequence. According to sequence convergence theorems we can prove that the sequence is increasing and bounded above. Increasing property is trivial because the series has absolute values.

$$|X^{(i)}| < c = \max(\{|X_0|, |X_1|, |X_2|\}) \forall i$$

$$|P^{(i)}| < q = \max(\{|\gamma_0|, |\gamma_1|, |\gamma_2|\}) \forall i$$

$$|Pr(i)| < q^{i-1} \forall i$$

$$\sum_{i=1}^n |X^{(i)} Pr(i)| \leq \sum_{i=1}^n cq^{i-1}$$

The right hand side of the inequality shows an infinite geometric progression with the common-ratio  $q$  and the initial term  $c$ . Since,  $q < 1$  this series converges to  $\frac{c}{1-q}$ .

$$\sum_{i=1}^{\infty} |X^{(i)} Pr(i)| \leq \frac{c}{1-q}.$$

$$\sum_{i=1}^n |X^{(i)} Pr(i)| \leq \frac{c}{1-q} \quad \forall n.$$

The sequence  $\sum_{i=1}^n |X^{(i)} Pr(i)|$  is bounded above. Hence The series,

$\sum_{i=1}^{\infty} |X^{(i)} Pr(i)|$  converges. The absolute convergence implies the convergence of the original series as well.  $\square$

*Proof.* (Theorem VII.1)

Now we can proceed with our optimality proof. We first have a general strategy  $Sr$  and  $Sr = \sum_{i=1}^{\infty} X^{(i)} Pr(i)$ . The value  $Sr$  exists as a real number by lemma VII.2. For the sake of argument

let  $\exists j$  s.t.  $X^{(j)} = X_2$ . Now define  $Sr_0$  to be the long-run pay-off achieved by changing the scissor move at iteration  $j$ , to a rock move. Also define  $Sr_1$  to be the long-run pay-off achieved by changing the scissor move at iteration  $j$  to a paper move. If we prove that  $Sr_0 - Sr < 0$  and  $Sr_1 - Sr < 0$  then it is evident that playing scissors at any iteration is optimal.

- Case 1: Proof of  $Sr_0 - Sr < 0$

It is clear that terms in the both series appear before  $j^{th}$  term cancels off when finding the difference. Let  $j^{th}$  term of the series  $Sr$  and  $Sr_0$  are  $Sr(j)$  and  $Sr_0(j)$  respectively.

$$Sr_0 - Sr = \sum_{i=j}^{\infty} Sr_0(i) - Sr(i).$$

$Sr_0(j) - Sr(j) = Pr(j)(X_0 - X_2)$ . Let  $M = \sum_{i=j+1}^{\infty} Sr(i)$ . With the change of move in  $j^{th}$  iteration, for every product  $Pr(i), i > j$   $P^j$  is changed from  $\gamma_2$  to  $\gamma_0$ .  $\sum_{i=j+1}^{\infty} Sr_0(i) = \frac{\gamma_0}{\gamma_2} M$ .

$$\sum_{i=j+1}^{\infty} Sr_0(i) - Sr(i) = \frac{\gamma_0}{\gamma_2} M - M.$$

$$Sr_0 - Sr = Pr(j)(X_0 - X_2) + \frac{P_0}{P_2} M - M.$$

Let's substitute the known values the expression,

$$Pr(j)(\frac{-3}{5} - (-1)) + \frac{4}{5} \times M - M.$$

Now it is sufficient to prove that,  $Pr(j)\frac{2}{5} + M < 0$

$$\iff |M| > Pr(j)\frac{2}{5}. (M < 0)$$

By dividing both sides of the inequality by  $Pr(j)$ , Left hand side of the inequality becomes,

$$M' = \sum_{i=j+1}^{\infty} (\prod_{k=j}^{i-1} P^{(k)}) X^{(i)}$$

$\iff |M'| > \frac{2}{5}$ . Now we find a lower bound for  $M'$  and prove that it is always greater than  $\frac{2}{5}$ . To find the lower bound we substitute the least absolute value among  $\{X_0, X_1, X_2\}$  and  $\{\gamma_0, \gamma_1, \gamma_2\}$  which would be  $\frac{3}{5}$  and  $\frac{2}{5}$  respectively to all the  $X^{(i)}$ 's and  $P^{(i)}$ 's. So,  $M'$  reduces to the geometric progression,

$\frac{3}{5}(\frac{2}{5} + \frac{2}{5} + \dots)$ . Which converges to  $\frac{2}{5}$ . The finite summation for any number of terms in  $|M'|$  is always  $< \frac{2}{5}$ . Now we can see that this lower bound is not sufficient to prove that  $|M'| > \frac{2}{5}$  for sufficiently large number of terms in the series

However, we know that the above substituted values do not show an existing path for a rational player. To arrive at an existing path we have to change at least one substituted value of  $X^{(i)}$ 's or  $P^{(i)}$ 's. Let's say this substitution increases the converged value by  $\epsilon$ . According to series convergence theorems we can also find  $\epsilon' < \epsilon$  s.t  $|M'| > \frac{2}{5} - \epsilon'$  for a finite summation of sufficiently large number of terms. After the new substitution,  $|M'|$  becomes  $M'' = |M'| + \epsilon$ . (the convergence can be proven)

$$|M'| + \epsilon > \frac{2}{5} - \epsilon' + \epsilon$$

$$\implies M'' > \frac{2}{5} \text{ for a sufficiently large number of terms.}$$

$$\implies |M| > \frac{2}{5} \text{ for a sufficiently large number of terms.}$$

- Case 2: Proof of  $Sr_1 - Sr < 0$

After substitution Case 2 reduces to prove that

$Sr_1 - Sr = Pr(j)(X_1 - X_2) + \frac{\gamma_1}{\gamma_2} M - M < 0$ . Now let's substitute the values to the expression it is sufficient to prove that .

$$Pr(j)(\frac{-2}{5}) + M < 0 \text{ which is obvious because } M < 0.$$

According to the best strategy of the rational player which is proven above, the optimal expected pay-off in the long-run for the rational player is  $\frac{-3}{5}$ .

Now we can prove the best strategy for an altruistic player. In this scenario we have two altruistic players and one Byzantine player. We have to consider the pay-offs for each move of the Byzantine player. We are going to prove that

Byzantine playing rock at each iteration results in the worst outcome for an altruistic player.

Values for  $\gamma_0$ ,  $\gamma_1$  and  $\gamma_2$  become  $\frac{14}{25}$ ,  $\frac{18}{25}$  and  $\frac{18}{25}$  respectively.  $X_0$ ,  $X_1$  and  $X_2$  become  $-0.4$ ,  $0.4$  and  $0$  respectively.

Similar to the previous proof, we can name the general expression for the cumulative expected pay off in long-run as  $Sr$ . The expression results in modifying rock move to paper move and scissor move are named as  $Sr_1$  and  $Sr_2$  respectively.  $Sr_1$  and  $Sr_2$  also exist by lemma VII.2. What we have to prove is  $Sr_1 - Sr > 0$  and  $Sr_2 - Sr > 0$ .

- Case 1: Proof of  $Sr_1 - Sr > 0$  After similar simplification to previous proof, It is left to prove that  $0.8Pr(j) + \frac{4}{14}M > 0$ . Here, each  $Pr(j)$  and  $M$  have same semantics to previous proof. If  $M > 0$  the result is trivial. We have to check the result only for the case where  $M < 0$ . We are going to find an upper bound for  $|M|$  when  $M < 0$ . This upper bound occurs when each term of  $M$  is negative. For a term of  $M$  to be negative, the rock strategy must be followed, because expected pay-offs of other two strategies are non-negative. To result in an upper bound each term in  $M$  have to be bounded above. We bound each term of  $M$  by substituting  $X^{(i)}$  with  $X_0 = -0.4$  and substituting  $P^{(i)}$  with  $\frac{18}{25}$  which is the maximum among  $\{\gamma_0, \gamma_1, \gamma_2\}$ . After the substitution, and dividing both sides by  $Pr(j)$  we have  $M' = \frac{M}{Pr(j)}$ .  $|M'| = 0.4(\frac{18}{25} + \frac{18^2}{25^2} + \dots)$  This converges to  $\frac{36}{35}$ . We have to prove that  $\frac{4}{14}|M'| < 0.8$ . By substituting the value of  $M'$  we get  $\frac{72}{245} < 0.8$ .  
 $\Rightarrow 0.8Pr(j) + \frac{4}{14}|M| > 0$ .
- Case 2: Proof of  $Sr_2 - Sr > 0$  It is left to prove that  $0.4Pr(j) + \frac{4}{14}M > 0$ . We can use the same upper bound for  $|M|$  as in Case 1 and the proof reduces to proving,  $\frac{72}{245} < 0.4$ .

The Byzantine player chooses the worst strategy for the altruistic player, which is proven above. The optimal expected pay-off in the long-run for the altruistic player is  $\frac{-10}{11}$ . Since  $\frac{-10}{11} \geq \frac{-5}{3}$  the altruistic strategy  $\langle \frac{1}{5}, \frac{1}{5}, \frac{3}{5} \rangle$  is a Nash-equilibrium when  $p_1$  is Byzantine.  $\square$

Now we can prove that the long-run expected pay-off  $V_i(init, t)$  and  $U_i(init, t)$  (notation in section III) also converge. Let  $s^{(1)}, s^{(2)} \dots$  denote the global state sequence visited during a game execution. If we fix a sequence of valid choices of actions by Byzantine and rational players we obtain a sub graph of the global graph. For example, a valid action choice sequence can be  $\langle (rock, paper), (rock, paper), \dots, (rock, paper) \rangle$  where we have one Byzantine player and one rational player.

**Definition VII.1.**  $Te = \{s | s \in S, \forall a \in A, \forall s' \in S T(s, a) \neq s'\}$

Formally, if a game terminates,  $P(\exists is.ts(i) \in Te) \rightarrow 1$ . Following theorem introduces the condition in global state graph to assure  $P(\exists is.ts(i) \in Te) \rightarrow 1$ . This condition

will also assure that the cumulative expected pay-off for any rational player converges.

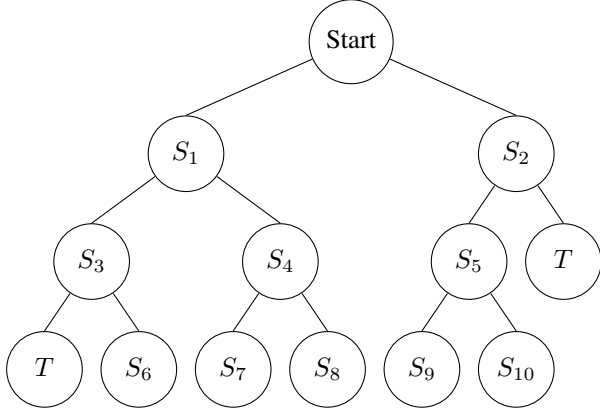
**Theorem VII.3.** *If for any sequence of valid action choices of Byzantine and rational players, the corresponding sub graph contains a path from each initial state to a terminal state then  $P(\exists is.ts(i) \in Te) \rightarrow 1$*

*Proof.* (Theorem VII.3) For a given sub graph corresponding to a sequence of valid action choices of Byzantine and rational players, there is a loop less longest path from an initial state to a terminal state. Let  $N$  be the length of longest of such paths among all the sub graphs corresponding to valid action choices. The minimum probability of reaching a terminal state occurs in the maximal length path.  $N$  represents the length of this path. Let's name the minimal probability for a transition as  $p_{min} > 0$ . The minimum probability exists because, the number of probabilistic transitions is finite. The probability of the given  $N$  length path ( $p_{min}^N$ ) is a lower bound for reaching any terminal state within  $N$  steps. We can extend this to reaching any terminal state within  $2N$  steps to  $p_{min}^N(1 + (1 - p_{min}^N))$ . Reaching a terminal state within  $iN$  steps is  $p_{min}^N(\sum_{k=0}^{i-1} (1 - p_{min}^N)^k)$  which becomes an infinite geometric series when  $i$  tends to  $\infty$ . This converges to  $p_{min}^N \times \frac{1}{1 - (1 - p_{min}^N)} = p_{min}^N \times \frac{1}{p_{min}^N} = 1$ . Since a series which is a lower bound to termination probability converges to probability 1, and the total probability which is an upper bound to the termination probability converges to probability 1. The actual termination probability also converges to probability 1 by sandwich theorem.  $\square$

**Theorem VII.4.** *Let  $i$  is an index for a non-Byzantine player according to section II. Let  $init \in I$ .  $V_i$  and  $U_i$  are as defined in section III. The sequences  $V_i(init, t)$  and  $U_i(init, t)$  converge when  $t \rightarrow \infty$ .*

*Proof.* We first prove that, cumulative expected pay-off under valid choice of actions of Byzantine and rational players converge. We should write a general expression to represent the cumulative expected pay-off. We denote the global action for the  $t^{th}$  step by the variable  $x_t$ .

To visualize the strategy, we present a tree diagram. The tree diagram represents the reachable global states with the actions of the altruistic players when the actions of Byzantine players and rational players are fixed. Probabilities for each branch in the tree diagram are the product of probabilities for each altruistic action which lead to the resulting global state. In other words, branch probability is a joint probability.



Node *Start* is in level 0, node  $S_1$  and  $S_2$  are in level 1 and so on. In the above diagram, first level containing a terminal state is found 2 steps after the starting state and one step after finding the first terminal state, second level containing the terminal state is found. Let's call  $n_i$  is the difference (in levels) between  $i^{th}$  level with a terminal state and  $i-1^{th}$  level with a terminal state for  $i > 1$   $n_1$  is the level number of the first level containing a terminal state.

Consider the minimum probability of reaching the first terminal state from the starting state which results in the maximum probability of the continuation of the game. Hence the maximum restarting probability after reaching the first terminal state is  $(1 - p_{min}^{n_1})$ . Let  $h_{max} = \max_s |SH_k(s)|$ . Cumulative pay-off value until the first level of termination can be bounded above by,

$n_1 \times h_{max} \times (1 - p_{min}^{n_1})$  Cumulative pay-off value from the first level of termination to the second level of termination can be bounded above by,

$n_2 \times h_{max} \times (1 - p_{min}^{n_1})(1 - p_{min}^{n_2})$ .

Cumulative pay-off value from the  $i-1^{th}$  level of termination to the  $i^{th}$  level of termination can be bounded above by,

$n_i \times h_{max} \times \prod_{j=1}^i (1 - p_{min}^{n_j})$

The formula for the upper bound of the cumulative pay-off value can be written as,

$\sum_{i=1}^{\infty} (n_i \times h_{max} \times \prod_{j=1}^i (1 - p_{min}^{n_j}))$  Since  $n_i, n_j$  are strategy dependant, we substitute  $N$  in the place of  $n_i$  and  $n_j$  to arrive at an upper bound which matches all the strategies.

$\sum_{i=1}^{\infty} (N \times h_{max} \times (1 - p_{min}^N)^i)$

By using the infinite summation for geometric series, the upper bound can be written as,

$$\frac{N h_{max}}{p_{min}^N}$$

Note that, in the formula for a general strategy we have to substitute the exact expected pay-off value in the place of  $n_i \times h_{max}$ . We can see that by substituting  $h_{max}$  we have also found out that the series for the absolute values of the pay-offs is also bounded above.

Since the absolute pay-off series is a summation of positive values, the summation is an increasing function. This means absolute pay-off series is convergent which implies the actual pay-off series is also convergent.

We proved above, cumulative expected pay-off for any altruistic strategy (Byzantines and rationals are fixed) converges in the long run. The values calculated for  $V_i$  and  $U_i$  should also

converge. □

## VIII. CONCLUSION

In this article, we presented a framework for probabilistic BAR systems. We introduced an algorithm to verify Nash-equilibrium properties in concurrent games with perfect and imperfect information at the level of global state knowledge. We performed our algorithm on two case studies, the 3-player Rock-Paper-Scissors example and Halpern et al.'s Secret Sharing protocol. Some results are interesting as they illustrate the necessity of a Byzantine player in a game to "force" the rational player to behave altruistically notably in the Rock-Paper-Scissors example. This opens a new door for Byzantine players, which are currently only considered as misconfigured or corrupted agents. Our algorithm is guaranteed to terminate (due to the maximum number of iterations) but the convergence may be not guaranteed if  $V_i$  and  $U_i$  converge to the same value. The analysis of this case is left to future work.

## REFERENCES

- [1] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J. Martin, and C. Porth, "BAR Fault Tolerance for Cooperative Services," in *Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005, SOSP 2005, Brighton, UK, October 23-26, 2005*, 2005, pp. 45–58.
- [2] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin, "BAR Gossip," in *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA, 2006*, pp. 191–204.
- [3] F. Mari, I. Melatti, I. Salvo, E. Tronci, L. Alvisi, A. Clement, and H. C. Li, "Model Checking Nash Equilibria in MAD Distributed Systems," in *Formal Methods in Computer-Aided Design, FMCAD 2008, Portland, Oregon, USA, 17-20 November 2008*, 2008, pp. 1–8.
- [4] R. Brenguier, "PRALINE: A tool for computing nash equilibria in concurrent games," in *Proc. 25th Computer Aided Verification*, ser. LNCS, vol. 8044. Springer, 2013, pp. 890–895.
- [5] T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games," in *Proc. 19th Tools and Algorithms for the Construction and Analysis of Systems*, ser. LNCS, vol. 7795. Springer, 2013, pp. 185–191.
- [6] R. Pass and J. Halpern, "Game Theory with Costly Computation: Formulation and Application to Protocol Security," in *Proceedings of the Behavioral and Quantitative Game Theory - Conference on Future Directions, BQGT '10, Newport Beach, California, USA, May 14-16, 2010*, 2010.
- [7] I. Abraham, D. Dolev, R. Gonen, and J. Halpern, "Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation," in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*. ACM, 2006, pp. 53–62.
- [8] F. Dileepa, N. Dong, C. Jegourel, and J. S. Dong, "Verification of nash-equilibrium for probabilistic bar systems-full-version." [Online]. Available: <https://github.com/dileepa123/ICECCS>
- [9] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [10] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation: Extended abstract," in *Proc. 36th Annual ACM Symposium on Theory of Computing*. ACM, 2004, pp. 623–632.