

# An observation on pure strategies in Security Games

Marek Adrian

Department of Applied Computer Science  
AGH University of Science and Technology  
Al. A. Mickiewicza 30, 30-059 Kraków, Poland  
Email: madrian@agh.edu.pl

Gianluigi Greco

DeMaCS Dept., University of Calabria  
Via Pietro Bucci, 87036 Arcavacata,  
Rende (CS), Italy  
Email: ggreco@mat.unical.it

**Abstract**—Security Games have been used in several different fields to randomise the division of limited resources and thus maximise the possibility of securing a set of targets. For this very practical purpose it is natural to consider primarily mixed strategies, but such focus omits some theoretical properties of the games discussed. In this paper we discuss the existence and properties of pure Nash equilibria in security games. We give an overview of the basic observations that can be made in this setting. We also recognize an interesting problem in a case with multiple players playing a security game asynchronously, propose an algorithm for finding a strategy for any given player in the mentioned case and prove that the strategy profile resulting from the algorithm is in fact a Nash equilibrium and, even stronger, a subgame perfect equilibrium. We think that these findings are a nice supplement of the practical approach to Security Games and allow to form new research questions.

## I. INTRODUCTION

Since its conception in the previous century, Game Theory provided a language which has been used to discuss, among other things, how businesses compete on a given market, how to model predator-pray interactions in the animal kingdom and, most famously, how to behave during an interrogation. This should come at no surprise as, at its core, game theory describes conflict between autonomous entities and such beings can be recognized in almost any setting.

One of the most basic concepts that was necessary to define in this field from the very beginning, is how to recognize whether any decision made by an entity is good or not. Typically it is decided by considering the potential outcome, which is provided by knowing the decision process of all of the players, and testing whether it has a desired set of properties.

One of the most known descriptions of a good strategy profile is the Nash equilibrium [1]. While it has its own drawbacks, it provides a reasonable set of assumptions on the behaviour of the players and always exists in finite games with mixed strategies.

A mixed strategy is when the decision process of a player is given by a probabilistic function over the possible actions, and thus is great to describe the uncertainty in decision making. In contrast we have pure strategies, where a player chooses one action. There is no guarantee that there is a Nash equilibrium consisting of only of pure strategies, which makes it a very interesting decision question, that has been studied for several classes of games [2] [3].

An interesting type of game, for which pure strategies were not considered, are the so called security games. Based on an

idea by Stackelberg [4], these games divide the players into two groups, of which one declares their strategies before the other, and tries to find a Nash equilibrium in such setting. This approach has been successfully applied in several systems, like the security checkpoint schedule in LAX airport [5], planning US Air Marshals flight security patterns [6], preventing poaching [7] and other cases. While it should be obvious why limiting yourself to just pure strategies in security games is not the best thing to do from a practical point of view, we found the theoretical properties of this model to be interesting and this paper, which is based on a PhD thesis [8], provides some insights to the approach.

The structure of this paper is the following: In section 2 we provide the basic definitions needed for the discussion and observations that can be made about pure strategies in security games. In next section we discuss deeper a specific case where we have multiple defenders playing asynchronously and recognize which game theory concepts should be used to find the strategies in that setting. Section 4 contains the algorithm for choosing a strategy for each player and the proof that the resulting strategy profile is a subgame perfect equilibrium. In the final section we will briefly summarize our observations and recognize further possible areas of inquiry.

## II. BASIC DEFINITIONS AND OBSERVATIONS

As a *game in normal form* we recognise a tuple  $(N, A, u)$  where:  $N$  is a finite set of players, indexed by  $i$ ;  $A = A_1 \times \dots \times A_n$  where  $A_i$  is a finite set of actions available to player  $i$ . Each vector  $(a_1, \dots, a_n)$  in  $A$  is called an *action profile*;  $u = (u_1, \dots, u_n)$ , where  $u_i : A \rightarrow \mathbb{R}$  is a *valuating function* for player  $i$ . In our cases we will assume that the goal of each player is to get the largest possible value from their valuating function. Moreover in the games we will be discussing any action will be corresponding to choosing an object to protect and thus we sometimes use the expressions *committing to an action* and *picking an object* interchangeably, hopefully not causing too much confusion. The assignment of probabilities by a player to his set of available actions is called a *strategy* and the decision on the probabilities is called *committing to a strategy*. If the assignment gives one of the actions probability 1 then it is called a *pure strategy*. Any other strategy is called *mixed*. A *strategy profile* is a vector containing strategies of all players in a given game. To formalize the concepts of good strategies we define a *best response* for the player  $i$  to an action

vector  $a_{-i}$  (an action vector without the  $i$ -th position) is an action  $a \in A_i$  such that for all  $a'$  in  $A_i$  we have  $u_i(a_{-i}, a) \geq u_i(a_{-i}, a')$ . A strategy  $a = (a_1, \dots, a_n)$  is a *Nash equilibrium* iff for all  $i$  in  $\{1, \dots, n\}$   $a_i$  is a best response to  $a_{-i}$ .

Now we can move on to define security games, in which we divide the players in to two groups, one of which declares their strategies before the other. We call the first group the defenders and the second group is called the attackers. In any case an action done by a defender is interpreted as defending a specific object, corridor, monument etc. and an action chosen by the attacker is assaulting it. For it to be a security game, the valuation function has to have an additional property: if the attacker chose to attack an undefended target his valuation function will be higher than if the target was defended. Symmetrically the valuation function for the defender should be worse when an undefended object has been attacked, then if a defended object has been attacked.

While the properties of the valuation function play a crucial role in finding mixed strategies in Security Games, there are not that important when we assume that only pure strategies are available to the players. In the case of pure strategies in security games it is sufficient to just consider the strategies of the defenders. Now let us think what will happen, if all of the defenders have to commit to a strategy at the same time. Obviously, they want their most valued object to be protected and they have no way to coordinate with other players. If for each of them the most valuable object is different, then we will have a Nash equilibrium.

If two players value one object the most we have an interesting situation: on one hand if they are rational they should pick the most valued object, but that will lead to a strategy profile that is not a Nash equilibrium, as if only one of them would switch to his second best object, he would increase his valuation function. On the other there is no rational way for any of those players to make a different decision as they are risking lesser value if both of them choose their second best option. So we would have a situation where there may exist a pure Nash equilibrium, but there is no way for the players to achieve it. This problem could disappear, if the defenders themselves played in a given order, but it may not be the case, and will be the topic of our next inquiry.

### III. MULTIPLE DEFENDERS IN AN ASYNCHRONOUS GAME

Let us consider a security game with  $n$  defenders and one attacker. Each action of a defender consists of picking an object to defend. The defenders commit to their pure strategies in a given order. The valuation function for each player is given as the sum of all values of the objects picked by all of the players. This model describes a sequential game which can be described as a game in Extended Form. Full formal definitions of an Extended Form game, Nash equilibrium and a sub-game perfect equilibrium in such games can be found in handbooks like [9] or [10]) We will describe the basic intuitions behind those concepts.

We can represent a game in Extended Form as a tree in which: each vertex represents the state of the game at the

moment, the root being the game before any move was made, and each leaf describing each possible outcome of the game, each edge represents an action that the current player can choose and connects the vertex corresponding to the game state before that action to the vertex with the game state after that action. With this representation any sub-tree that starts in a vertex and consists of all the edges and vertices below is also a game and is called a *sub-game*.

Any game in extended form can be translated into normal form and thus we can use the definitions of Nash equilibrium and best response in this context. There is a problem however, as the Nash equilibrium does not have to be optimal on sub-games. A *sub-game perfect equilibrium* is a Nash equilibrium that is also a Nash equilibrium on all of their sub-games.

Now if we have the complete game tree, it is easy to see that we can find a best response for any player simply by backtracking the expected results from the leaves to the current situation. This is unfeasible, as the whole game tree will grow exponentially with respect to the number of players and possible actions. To get rid of this problem, instead of trying to find a whole strategy, we will try to identify a good move and argue that there exists a sub-game perfect equilibrium in which this was the best response.

Consider a game  $G$  with the set of actions  $A$  and a strategy profile  $s$  for  $G$ . A sequence  $(a_1, \dots, a_n) \in A^n$  is a *result* of strategy  $s$  if, and only if starting in the root of the game tree and moving down an edge only if it is the action indicated by the strategy  $s$ , the actions the edges traveled through form the sequence  $(a_1, \dots, a_n)$ .

We say a sequence of actions  $(a_1, \dots, a_n)$  is called *reasonable* if there exists a strategy profile that is a sub-game perfect equilibrium, such that  $(a_1, \dots, a_n)$  is a result of  $s$ .

Thus we can simplify our problem and instead of finding a strategy profile which should describe actions taken in any possible situation, just find an sequence of actions and argue that they are a result of a good strategy profile.

## IV. MAIN RESULT

We will present now the algorithm for finding good move for each player. The algorithm in itself is fairly simple and easily works in polynomial time.

### A. Algorithms for decisions

#### The basic algorithm

**Input:**  $A$  - set of available actions;  $V$  - the valuation matrix;  $i$  - the index of the player making the decision;

**Output:**  $(a_i, \dots, a_n)$  the predicted choices of actions for players  $i$  to  $n$ .

- 1) Delete all columns for actions that have already been chosen.
- 2) Define  $k$  as the number of rows in the matrix.
- 3) Find in the last row the column in which there is the most valuable object for the  $k$ -th player (if more than one pick at random).
- 4) Mark this object as  $a_k$ .
- 5) Remove the last row from the matrix.

6) Repeat steps 1-4 until  $a_i$  is defined.

### The modified algorithm

**Input:**  $A$  - set of available actions;

$V$  - the valuation matrix;

$i$  - the index of the player making the decision;

$(a_1, \dots, a_n)$  - sequence of choices of actions predicted by the original algorithm;

**Output:**  $(a'_i, \dots, a'_n)$  the predicted choices of actions for players  $i$  to  $n$ .

- 1) Delete all columns for objects that have already been chosen.
- 2) Define  $k$  as the number of rows in the matrix.
- 3) Find in the last row the column in which is the most valuable object for the  $k$ -th player (if more than one and  $a_k$  is available pick  $a_k$  else pick at random).
- 4) Mark this object as  $a'_k$ .
- 5) Remove the last row from the matrix.
- 6) Repeat steps 1-4 until  $a'_i$  is defined.

### B. Proofs of reasonability

To prove that the sequence provided by the basic algorithm is reasonable we will use the following lemma, which shows that if we have a sequence predicted by the basic algorithm and we remove one object from the set of possible objects, then using the modified algorithm for a player will have an output which will differ from the original output at most at one choice.

*Lemma 4.1:* Let  $(a_1, \dots, a_n)$  be the result of using the basic algorithm on the game  $G$  with the set of actions  $A$ . By running the modified algorithm for the game  $G$ , sequence  $(a_1, \dots, a_n)$  and set of objects  $A \setminus \{a\}$ , where  $a \in A$ , will give a sequence  $(a'_1, \dots, a'_n)$  which will differ from  $(a_1, \dots, a_n)$  in at most one element and only if  $a \in \{a_1, \dots, a_n\}$ .

Case 1: First let us consider the case in which  $a \notin \{a_1, \dots, a_n\}$ . In this case the resulting sequence will be identical to the original. As in step 4 of the modified algorithm for the  $k$ -player the action  $a_k$  will still be available and the valuation of objects has not changed, it will be chosen by the  $k$ -th player.

Case 2: Now consider the case in which  $a \in \{a_1, \dots, a_n\}$ . Let us assume that  $a = a_k$  for some  $1 \leq k \leq n$ . Since none of the values have changed and the players from  $k + 1$  to  $n$  will still have their previous choices available, the algorithm will pick those objects. Of course player  $k$  cannot choose  $a_k$  because it is not in the set of possible actions. The modified algorithm finds a new object for him which we will mark as  $a'$ . It cannot be that  $a' = a_i$  for  $i > k$  because these objects are already unavailable for the algorithm by now. If  $a' \neq a_i$  for all  $i < k$  then this will be the only change result of the algorithm, because then all of the objects the modified algorithm has to pick in case of ties are available as in case 1. If  $a' = a_i$  for some  $i < k$ , then it will be assigned to player  $k$ , but we have a problem with the assignment for player  $a_i$ . Running the modified algorithm for players from  $i + 1$  to  $k - 1$  will give the same result as the original, by the same reasoning as before. For player  $i$  we can repeat the same reasoning as

we did for the player  $k$ . As in each such repetition the index will get smaller and the sequence is finite, such replacement will happen only a finite number of times and will result in a sequence which differs only in one element from the original sequence.

*Theorem 4.2:* Let  $(a_1, \dots, a_n)$  be the result of using the algorithm on the game  $G$  with the set of objects  $A$ . Then  $(a_1, \dots, a_n)$  is reasonable.

We will prove this by assigning a move each vertex in the game tree and arguing that we can construct a sub-game perfect equilibrium for the game  $G$  for which  $(a_1, \dots, a_n)$  is the result. The proof goes by induction on the number of players. The case of  $n = 1$  is trivial.

$n = 2$ : We have one vertex corresponding to the decision of player 1 from which descent  $m$  edges corresponding to all possible actions for player 1. We put  $a_2$  on all vertices of player 2 except the one connected to the edge  $a_2$ . There we can use the modified algorithm on the sequence  $(a_1, a_2)$  and the set of objects  $A \setminus \{a_2\}$  to find one to put on this vertex. We put  $a_1$  on the root. It should be easy to see that this assignment will produce a sub-game perfect equilibrium with the sequence  $(a_1, a_2)$  as a result.

$n - 1 \Rightarrow n$ : Now we assume that we can assign moves tree for any game  $G$  with  $n - 1$  players a given set of objects  $A$  which constructs a sub-game perfect equilibrium and a sequence given by the algorithm is the result. We will show how to use this to assign moves in a tree for any game with  $n$  players and a sequence  $(a_1, \dots, a_n)$  given by the algorithm. We start from a tree for the game  $G$  with all vertices empty. For every vertex connected to the root we will run the modified algorithm on the game  $G$  without player 1, sequence  $(a_2, \dots, a_n)$ , and set of objects  $A \setminus \{a\}$ , where  $a$  is the label of the edge between this vertex and the root. By our inductive assumption, we can construct a full strategy on the subtree starting from that vertex, which has the desired properties and has  $(a'_2, \dots, a'_n)$ , given by the modified algorithm, as a result. We can see that by discarding the object  $a$  for this whole subtree we can be sure that, as long as we pick the proper object for the root, the whole strategy will stay a perfect subgame equilibrium. What is left is to show that there are no better actions to put at the root than  $a_1$ . Consider first edges  $a$  which are not in the set  $\{a_2, \dots, a_n\}$ . If player one was to pick one of them the result of playing the subtree under that edge, by the lemma, is exactly the sequence  $(a_2, \dots, a_n)$ , so it only could be beneficial for him if  $v_1^a > v_1^{a_1}$  which is contrary to the way we picked  $a_1$ . Suppose now that player 1 could benefit from committing to an action  $a$  from the set  $\{a_2, \dots, a_n\}$ . By the lemma the resulting sequence  $(a, a'_2, \dots, a'_n)$  differs in at most one element from the sequence  $(a_1, \dots, a_n)$ . If it differs, then for it to be beneficial it had to be the case that this one action has a greater value for player 1 than  $a_1$ , which is in contrary with the way  $a_1$  was chosen. So there is no action which grants a better result for player 1 than choosing  $a_1$ . We put  $a_1$  on the root getting a proper assignment to the tree which can construct a perfect su-game equilibrium with the result  $(a_1, \dots, a_n)$  thus completing the construction.

With multiple preferred objects it could happen that the whole outcome of the game is very different from what the players predicted and in fact the outcome does not have to be a sub-game perfect equilibrium, which would undermine the validity of the reasonable move as a good strategy concept. The next theorem proves that no matter how often the players were wrong in their predictions the whole outcome will be in fact reasonable.

*Theorem 4.3:* Let  $G$  be a game with  $n$  players and the set  $A$  of available objects. Player 1 uses the basic algorithm to obtain the sequence  $(a_1^1, \dots, a_n^1)$  and picks  $a_1^1$ . Then player 2 uses the basic algorithm on the set  $A \setminus \{a_1^1\}$ , obtains the sequence  $(a_1^2, \dots, a_{n-1}^2)$  and commits to  $a_1^2$ . The following players continue in a similar fashion cutting the set of objects. Then the sequence  $(a_1^1, a_1^2, \dots, a_1^n)$  is reasonable.

The proof goes by induction on the number of players. The case  $n = 1$  is trivial.

$n = 2$ : As in the previous proof we have one vertex corresponding to the decision of player 1 from which descent  $m$  edges corresponding to all possible actions for player 1. We put  $a_1^1$  on the one vertex of player 1. We put  $a_1^2$  on all vertices of player 2 except the one connected to the edge  $a_1^1$ . We use the modified algorithm for the sequence  $(a_1^1, a_1^2)$  and the set of objects  $A \setminus \{a_1^1\}$  to find what to place on the last vertex. This strategy will have  $(a_1^1, a_1^2)$  as a result. As to show that the assignment can be used to construct a sub-game perfect equilibrium it suffices to notice that even if  $a_1^2 \neq a_2^1$  both must be equally valued by player 2 because the algorithm gave those two elements as a possible move of player 2 on two different occasions, while both those actions were available to the player.

$n - 1 \Rightarrow n$ : We assume that we can build a strategy tree for any game  $G$  with  $n - 1$  players and a given set of actions  $A$  which is *PSE* and the proper sequence is the result. To show the result for  $n$  players we start with a game tree with all vertices empty. For every vertex connected to the root we run the modified algorithm on the game  $G$  without player 1, sequence  $(a_1^2, \dots, a_1^n)$  and the set of actions  $A \setminus \{a_1^1\}$ , where  $a_1^1$  is the label of the edge between this vertex and the root. By the inductive assumption the sequence  $(a_1^2, \dots, a_1^n)$  is reasonable for the proper sub-game, so the result of the modified algorithm is also reasonable and a sub-game perfect equilibrium can be constructed on this subtree. It remains to argue that after putting  $a_1^1$  in the root the strategy we get a similar result. It is important to notice that the sequence  $(a_1^1, a_1^2, \dots, a_1^n)$  is a possible result of using the regular algorithm for the game  $G$  with  $n$  players and set of objects  $A$ . Thus we can use the lemma for all the subtrees. So we can use the exact same argument as in the proof of the previous theorem to show that player 1 cannot benefit from changing committing to another move than  $a_1^1$ .

We can notice that this proof provides more than just the answer to this specific case. First of we didn't explicitly stated if there always exists a pure Nash equilibrium in the case of simultaneous moves by the defenders. We can see that any sub-game perfect equilibrium provided by our algorithm will

remain a Nash equilibrium, if all of the moves are made at the same time and so we see that a pure Nash equilibrium always exists in this case. If we would like to consider a case in which some of the defenders have the resources to defend more than one object we can simulate that by adding copies of that player's valuations to the valuation matrix and dividing one defender with  $n$  resources into  $n$  players with one resource and identical valuations.

## V. SUMMARY

In this paper we have discussed the properties of pure strategies in security games. We recognized which part of the model can be omitted in this situation and which can be redefined to simplify the model. We introduced a situation in which the defenders pick actions in a sequential order, we proved that a pure strategy equilibrium exists in such setting, that we can find a move corresponding to such an equilibrium in polynomial time, and used this construction to prove the existence to a pure Nash equilibrium in the general case of synchronous play of the defenders, as well as when the defenders have more than one resource to use.

As for future directions, the problem discussed in this paper assumed no possibility of communication between the defenders and we think that any query in that direction could be interesting. Also we can see that the result of the game in our case could depend heavily on the order in which the players were able to commit to their strategy. As so, we think that that finding out how much trying to coordinate the players, via an additional player or otherwise, could affect the possible result, or even finding out exactly how many different results can be achieved from a given game in any ordering, could pose also an interesting challenge.

## REFERENCES

- [1] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.
- [2] G. Gottlob, G. Greco, and F. Scarcello, "Pure nash equilibria: hard and easy games," *JAIR*, vol. 24, pp. 357–406, 2005.
- [3] R. W. Rosenthal, "A class of games possessing pure-strategy nash equilibria," *Int. Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973.
- [4] H. Von Stackelberg, *Market structure and equilibrium*. Springer Science & Business Media, 2010.
- [5] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 125–132.
- [6] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe, "Iris-a tool for strategic security allocation in transportation networks," *AAMAS (Industry Track)*, pp. 37–44, 2009.
- [7] F. Fang, P. Stone, and M. Tambe, "When security games go green: Designing defender strategies to prevent poaching and illegal fishing," in *IJCAI*, 2015, pp. 2589–2595.
- [8] M. Adrian, "Pure strategies in security games," Ph.D. dissertation, UNICAL Università della Calabria, 2017.
- [9] K. Leyton-Brown and Y. Shoham, "Essentials of game theory: A concise multidisciplinary introduction," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 2, no. 1, pp. 1–88, 2008.
- [10] S. Tadelis, *Game theory: an introduction*. Princeton university press, 2013.