

# On Optimal AV System Strategies against Obfuscated Malware

Anshuman Singh\*, Bin Mai†, Arun Lakhotia\* and Andrew Walenstein\*

\*University of Louisiana at Lafayette, LA, USA

†Northwestern State University, Natchitoches, LA, USA

**Abstract**—Many Anti-Virus(AV) Systems are heterogeneous compositions of components, with each component specially tuned to work on a certain class of threat. Each component may have individually tunable parameters and different performance characteristics. No general theory is known for composing such components and assigning their individual parameters in order to ensure optimal resistance to attack. A particularly important question is posed by the possibility of obfuscated malware, which may fool the system into using different components. This paper introduces a framework for modeling composite AV Systems as classifiers wired together using selectors. It then uses game theory to analyze possible attacks. According to the game analysis, using a selector is beneficial only when the cost of developing obfuscated malware to game it is above a certain threshold. In this paper, we then *derive* the optimal configuration of detection components of an AV System such that the attacker’s use of obfuscation is rendered ineffective.

## I. INTRODUCTION

**M**ANY computer defense systems rely on multiple components that are composed into a single system that, in combination, is used to defend against attacks. For example, a mail server may pass incoming mail to multiple Anti-Virus(AV) products from different vendors before letting the mail through. And, at a finer level of granularity, a single AV product may also be composed of several discernible detector components. For example, a single product may include a component for matching cryptographic checksums, for ordinary signatures, for so-called "x-ray" scanning, for static behavior-based patterns, an emulation-based behavior matcher, and a run-time behavior matcher based on monitoring hooked system calls [1].

There are several important rationales for constructing heterogeneous AV Systems. First, it often is simply good software engineering practice to decompose large systems into smaller, well-defined components. Second, it may be the case that certain detector components are applicable to only certain inputs. For example, an AV system’s static behavior pattern matcher may be known to work well on only specific types of malicious files. Third, there may be important performance reasons for dividing the work up between components; in particular, certain components may incur much higher computational cost than others, so it is important to ensure that they are used in those situations in which they are most likely needed, and not on all files in general. Whatever the reasons for using multiple components, they must be wired together in a way that they work in coordination to perform the detection.

An essential piece is the logic used to select the inputs that the various components will work on, so that the computational costs are kept low and the components are used only on the appropriate inputs. Another issue to consider is the settings of tunable parameters for the components.

A critical concern is whether the system, as a whole, is made more resilient to attack by virtue of its combination of components and connection logic. A specific problem is caused by the possibility of using various anti-AV techniques, such as obfuscation, to game the selection of different classifiers. In particular, obfuscation may be used to fool a single detection component into making the wrong decision but, with selection logic added to the system, new obfuscation attacks are made possible directly on the selection logic. Thus new questions arise as to whether the compositions are more resistant than the individual components, and how to assign any detector parameters that are tunable. No generic framework or analysis method is known for answering such questions.

This paper proposes a modeling framework and analysis technique that can help begin answering such critical questions. The framework for modeling heterogeneous AV Systems treats them as a combination of classifiers connected together using probabilistic selectors. From such models, defense construction (AV Software setup) and attacks on them are treated as a game. For example, the virus-antivirus coevolution described by [2] can be modeled as a game in this framework. A game theoretic analysis can then be performed that can expose potential attack weaknesses. By setting up a game using variables in the models instead of actual constants, an abstract game model can be constructed. Though game theory has been applied in computer science in semantics of programming languages and logic systems(game semantics) [3], adversarial classification in KDD systems [4], and artificial intelligence [5], we apply it in a malware-anti-virus scenario and derive formally some interesting relationships among the parameters of the game.

Using a sample play of a game with a two-component AV System, the paper shows that interesting general characteristics of composite AV Systems can be extracted. Specifically, it characterizes the conditions in which the AV System as a whole is made weaker by the addition of a selector and specific classifier. More specifically, we found that, first, within our model setting, augmenting detection with a selector would not always benefit the AV System. The selector’s value can be fully realized only when the cost of obfuscating malware is above

a certain threshold; secondly, the AV System is always better off by configuring its classifiers so as to render the use of obfuscation in malware ineffective, and this can be achieved by decreasing the detection rate of the classifier designed for malware and increasing the detection rate of the classifier designed for the normal files. This implies that when cost of developing obfuscated malware is low and selection accuracy is high, the difference in detection rates of the classifiers should be low for optimal performance of the AV System.

The rest of the paper is organized as following: we give an overview of the basic concepts from game theory in Section 2. We then describe, in section 3, the design decisions used in modeling AV Systems. In section 4 we set up the Malware Author-AV System game and describe each play of the game as a configuration followed by payoff decision trees for each player. In section 5 we compute expected payoffs for the players and derive relationships between tunable parameters of the AV System and malware development cost. In Section 6, we discuss the implication of our results and conclude the paper.

## II. BACKGROUND

Game theory, a branch of applied mathematics, is useful for making decisions in situations where two or more rational decision makers have conflicting interests. Applications of game theory attempt to find equilibria in these games-the combination of the strategies for each agent in which none of the agents have incentive to change their strategy. This is an analytical tool that is especially valuable in analyzing situations where there are strategic interactions among multiple agents and each agent's behavior and consequences are intricately related to each other's.

A game is a situation of strategic interdependence that consists of a set of players, a set of strategies available to those players, and a specification of payoffs for each combination of strategies. The extensive and the normal forms are used to define noncooperative games-games in which the goal of each player is to achieve largest possible individual gain. A normal-form representation of a game is a specification of players' strategy spaces and payoff functions that is graphically represented as a 2-d matrix for a two player game. A *strategy space* for a player is the set of all strategies available to that player, where a strategy is a complete plan of action for every stage of the game, regardless of whether that stage actually arises in play. A *payoff function* for a player is a mapping from the cross-product of players' strategy spaces to that player's set of payoffs. These payoffs are the sum of benefits and costs to the player obtained by choosing a strategy. The reader can refer [6] for a thorough introduction to game theory.

## III. MODELING HETEROGENEOUS AV SYSTEMS

A simple AV System with a single detection component can be thought of, abstractly, as a classifier that classifies its inputs into one of possibly several categories. In this case, the inputs are potentially malicious programs, and the output classes might be, for example, clean, suspicious, and dirty. Classifiers such as these may be connected together in parallel so that

for any given input all classifiers are run, and the outputs are combined in some manner. An example is shown in Figure 1(a). In such configurations, well-understood analysis methods such as boosting can be found in the classifier literature [7].

For composite AV Systems such configurations are not desirable since not only is it too costly to run all classifiers on all inputs, it is frequently the case that certain classifiers are specialized to work only on certain subsets of the input space. The composite AV Systems can be modeled as a combination of classifiers connected together using probabilistic selectors (Figure 1(b)). The selector performs some form of lightweight scanning based on which it subjects the input to a specialized classifier. For example, most second generation AV scanners use as a selector a nearly exact identification method using cryptographic checksums ([1], p.437-8) and then based on selector's decision subject the suspected file to a particular algorithmic scanning method ([1], p.441) that can be considered as a specialized classifier.

As another example, consider the case of a normalizing detector for metamorphic malware similar to the one defined in [8]. Although the algorithm used is more efficient than semantics-based static normalization approaches, the normalization is likely most helpful only for a small number of files, so for performance reasons the normalizer is likely to be combined with a selector component that can quickly filter out the files that are highly unlikely to need the normalization. A fast statistical selector [9] might be used in combination with the normalizer. It selects whether the incoming file is likely to be metamorphic and in need of going through the normalization process described in [8]. In this way, the majority of files need not be scrutinized by the more heavyweight normalizer. This is a classic instance of a specialized classifier approach.

## IV. THE MALWARE AUTHOR-SECURITY ANALYST GAME

Malware authors always try to develop malware that evades detection by an AV System and the Security analyst will always try to come up with a design and configuration of an AV System that improves the detection rate. This situation can be modeled as a game. The players involved in this game are malware author (MA) and security analyst (SA). The strategies for MA are either to develop a low cost unobfuscated malware (UM) or the more expensive obfuscated or metamorphic malware (OM). The strategies for SA are either to use a single classifier (C architecture) or two classifiers with a selector (S2C architecture) for the AV System as discussed in Section III. We assume the strategies are pure for each player, though a game with mixed strategies can also be modeled similarly. The game can be described in normal form as in Figure 2. MA's strategies are given in rows and SA's in columns. Each play of the game is called a configuration and there are four such configurations: UM-C (UMC), OM-C (OMC), UM-S2C (UMS2C) and OM-S2C (OMS2C). The payoffs for each player are given in the pair for the corresponding play of the game where each player has chosen one of the strategies. The first element of the pair is the expected payoff for the Malware Author and the second element is the expected payoff for the Security Analyst. The components of the AV System

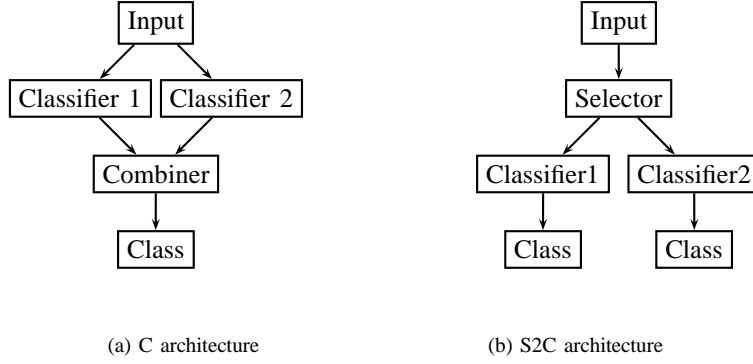


Fig. 1: Methods of composing multiple classifiers in an AV System

	<b>C</b>	<b>S2C</b>
<b>UM</b>	$(\pi_{UMC}^{MA}, \pi_{UMC}^{SA})$	$(\pi_{UMS2C}^{MA}, \pi_{UMS2C}^{SA})$
<b>OM</b>	$(\pi_{OMC}^{MA}, \pi_{OMC}^{SA})$	$(\pi_{OMS2C}^{MA}, \pi_{OMS2C}^{SA})$

Fig. 2: The MA-SA game in normal form

make classification and selection decisions based on which the payoffs can be computed for each player for each decision path. The costs and benefits to players for various outcomes and the parameters of the AV system components are described below.

When normal files are sent to the AV System, SA derives a positive utility of  $\nu$ . We assume that if AV System successfully detects the malware, she completely avoids any loss and MA gets  $\mu_l$  payoff ( $\mu_l > 0$  such that SA always has incentive to try to block an attack). If AV System fails to detect the malware, SA incurs a damage of  $d$  and MA obtains a payoff of  $\mu_h$  ( $\mu_h > \mu_l$ ). Once the AV System detects malware, regardless whether it is true positive or false positive, we assume SA would incur a cost of  $c$  for taking appropriate steps to protect itself. When MA notices that SA decides to configure the AV System such that it uses a Selector for pre-screening to choose between a lenient and stringent classifier, MA may attempt to develop obfuscated malware to make the Selector send its file to the lenient classifier. MA incurs an additional cost of  $\Delta$ , to develop obfuscated malware. The benefits and costs are summarized in Table I.

Input	Outcome	Benefits		Costs	
		SA	MA	SA	MA
Normal	Detected	$\nu$		$c$	
	Missed	$\nu$			
Malware (unobfuscated)	Detected		$\mu_l$	$c$	
	Missed		$\mu_h$	$d$	
Malware (obfuscated)	Detected		$\mu_l$	$c$	$\Delta$
	Missed		$\mu_h$	$d$	$\Delta$

TABLE I: Costs and benefits obtained by each agent for all possible outcomes

In the single classifier architecture (C) of the AV System,

let  $p_D$  denote the detection rate (true positive rate) of the classifier, i.e. the probability that classifier correctly detects MA's malware. Since the classifier can also give false positives when scanning through clean files, we denote the false positive rate by  $p_F$ . A classifier can be configured to operate at a specific combination of  $(p_D, p_F)$  values on its Receiver Operating Characteristics (ROC) curve, which specifies the permissible combinations for the device [10]. An ROC curve represents  $p_D$  as an increasing concave function of  $p_F$ . We assume that the ROC curve is given by the power function  $p_D = p_F^r$ , with  $0 < r < 1$ .

In S2C architecture of the AV System, for the selector (S) the probability of selecting a normal input file as normal is  $t_N$  and the probability of selecting input that is malware as malware is  $t_M$ . The file selected as normal is directed to the lenient classifier (LC) and the file selected as malware is directed to the stringent classifier (SC). The true positive rate and the false positive rate of the stringent classifier (SC) is  $p_D^S$  and  $p_F^S$  respectively. Similarly, the true positive rate and the false positive rate of the lenient classifier (LC) is  $p_D^L$  and  $p_F^L$  respectively.

Figures 3, 4, 5, 7, 6, 8 give the decision trees for each player for all the configurations of the game. Figure 4 gives the decision trees for SA for both UMC and OMC configuration since the payoffs remain the same. The outcomes 'Detected' and 'Missed' are represented by '+' and '-', respectively, in these figures.

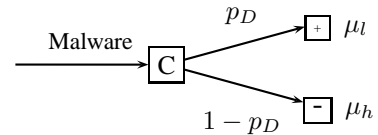


Fig. 3: MA's payoff in configuration UMC

## V. AV SYSTEM OPTIMAL PARAMETERS

We now analyze the different game configurations by computing expected payoffs for the players. Maximizing the expected payoffs under certain conditions can help tuning the AV System parameters for optimal detection rates. This kind

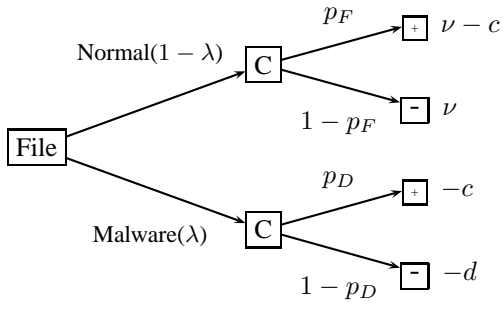


Fig. 4: SA's payoff in configuration UMC (OMC)

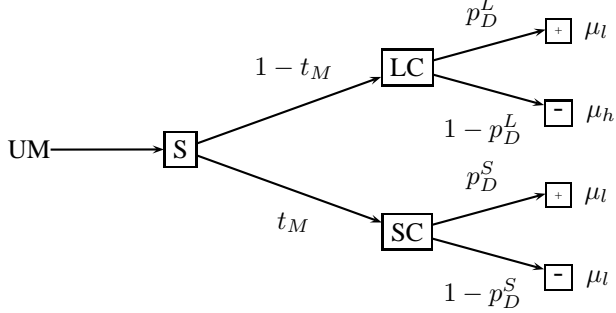


Fig. 5: MA's payoff in configuration UMS2C

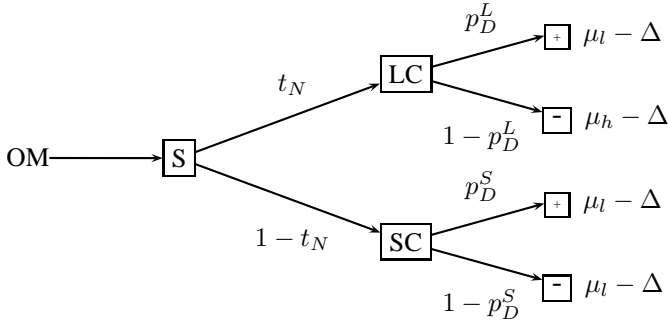


Fig. 6: MA's payoff in configuration OMS2C

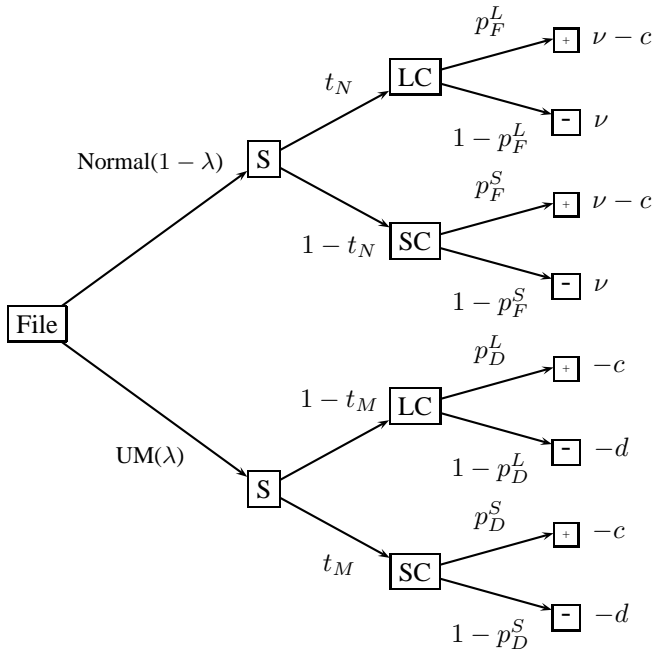


Fig. 7: SA's payoff in configuration UMS2C

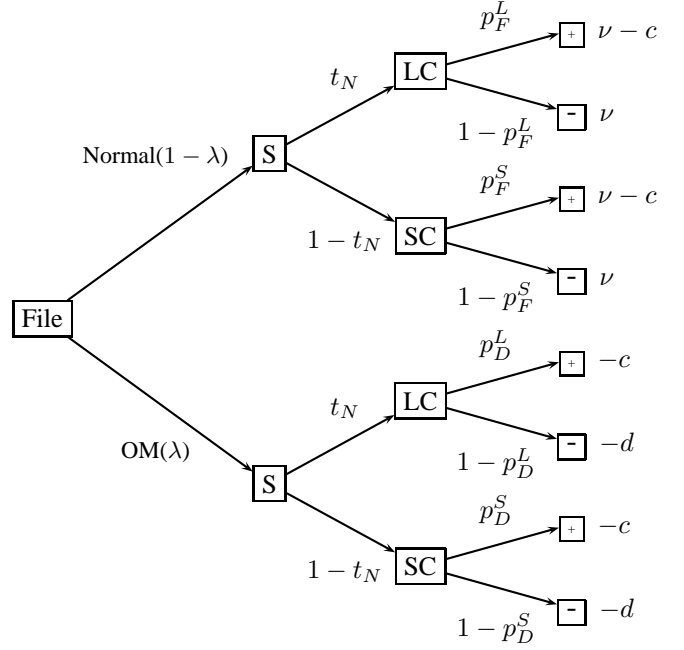


Fig. 8: SA's payoff in configuration OMS2C

of analysis helps in deriving interesting relationships between various parameters in the game. These relationships are based on formal and rigorous analysis instead of ad hoc heuristics. The variables used in decision trees and the expressions below are defined and explained in section IV and summarized in Table I.

#### A. The UMC Configuration

In this configuration SA chooses to use only one classifier and does not pre-screen incoming files, thus every file is sent through the same classifier device. When the input file is normal and the classifier C incorrectly classifies it as a malware, the SA will get benefit  $\nu$  for using the classifier and incur a cost  $c$  for incorrect classification of the input (see Table I). Thus, the payoff to SA will be  $\nu - c$  (see Figure 4). As the probability of an input file being clean or normal is  $1 - \lambda$  and the false positive rate of the classifier C is  $p_F$ , the weighted payoff to SA is  $(1 - \lambda)p_F(\nu - c)$ . The expected payoff for SA can be obtained by adding the weighted payoffs for all other paths in the decision tree. The expected payoff for SA in UMC configuration is:

$$\begin{aligned} \pi_{UMC}^{SA} &= (1 - \lambda)p_F(\nu - c) + (1 - \lambda)(1 - p_F)\nu + \\ &\quad \lambda p_D(-c) + \lambda(1 - p_D)\nu \\ &= \nu - (d + \nu)\lambda - c(1 - \lambda)p_F + (d - c)\lambda p_D \end{aligned}$$

SA can configure  $p_D$  to its optimal value by maximizing expected payoff for SA:

$$\frac{d(\pi_{UMC}^{SA})}{dp_D} = 0 \quad (1)$$

Solving (1) using  $p_D = p_F^r$  yields  $\bar{p}_D$ , the optimal value of  $p_D$ :

$$\bar{p}_D = \left[ \frac{c}{d - c} \cdot \frac{1 - \lambda}{\lambda} \cdot \frac{1}{r} \right]^{\frac{r}{r-1}} \quad (2)$$

If  $\alpha = (1 - \lambda)/\lambda$  is the ratio of normal files to malware,  $\delta = d/c$  is the damage to cost ratio and  $\xi = \alpha/(\delta - 1)$ , then (2) can also be written as:

$$\bar{p}_D = \left[ \frac{\xi}{r} \right]^{\frac{r}{r-1}}$$

### B. The UMS2C and OMS2C configuration

In OMS2C configuration, the expected payoff for MA can be computed from Figure 6 as:

$$\begin{aligned} \pi_{OMS2C}^{MA} &= (1 - t_N) \{ p_D^S (\mu_l - \Delta) + (1 - p_D^S) (\mu_h - \Delta) \} + \\ &\quad t_N \{ p_D^L (\mu_l - \Delta) + (1 - p_D^L) (\mu_h - \Delta) \} \\ &= t_N (\mu_h - \mu_l) (p_D^S - p_D^L) + p_D^S (\mu_l - \mu_h) + \mu_h - \Delta \end{aligned}$$

If  $\pi_{OMS2C}^{MA} \leq 0$ , MA will choose strategy UM. This occurs when

$$t_N (\mu_h - \mu_l) (p_D^S - p_D^L) + p_D^S (\mu_l - \mu_h) + \mu_h - \Delta \leq 0$$

and since  $\mu_l - \mu_h < 0$

$$\Delta \geq t_N (\mu_h - \mu_l) (p_D^S - p_D^L) + \mu_h \quad (3)$$

If condition (3) holds, then optimal values of  $p_D^S$  and  $p_D^L$  can be obtained by maximizing the expected payoff for SA in UMS2C configuration. The expected payoff for SA in UMS2C configuration can be computed from Figure 7 as:

$$\begin{aligned} \pi_{UMS2C}^{SA} &= (1 - \lambda) t_N \{ p_F^L (\nu - c) + (1 - p_F^L) \nu \} + \\ &\quad (1 - \lambda) (1 - t_N) \{ p_F^S (\nu - c) + (1 - p_F^S) \nu \} + \\ &\quad \lambda (1 - t_M) \{ (p_D^L (-c) + (1 - p_D^L) (-d)) \} + \\ &\quad \lambda t_M \{ p_D^S (-c) + (1 - p_D^S) (-d) \} \end{aligned}$$

Maximizing the above expression w.r.t.  $p_F^S$ , we obtain:

$$\bar{p}_F^S = \left[ \frac{c(1 - \lambda)(1 - t_N)}{(d - c)\lambda r t_M} \right]^{\frac{1}{r-1}}$$

It follows that:

$$\bar{p}_D^S = \left[ \frac{c}{d - c} \cdot \frac{1 - \lambda}{\lambda} \cdot \frac{1 - t_N}{t_M} \cdot \frac{1}{r} \right]^{\frac{r}{r-1}} \quad (4)$$

Similarly

$$\bar{p}_D^L = \left[ \frac{c}{d - c} \cdot \frac{1 - \lambda}{\lambda} \cdot \frac{t_N}{1 - t_M} \cdot \frac{1}{r} \right]^{\frac{r}{r-1}} \quad (5)$$

Using notation from section V-A, (4) and (5) can be written as

$$\bar{p}_D^S = \left[ \frac{\xi}{r} \cdot \frac{1 - t_N}{t_M} \right]^{\frac{r}{r-1}}$$

and

$$\bar{p}_D^L = \left[ \frac{\xi}{r} \cdot \frac{t_N}{1 - t_M} \right]^{\frac{r}{r-1}}$$

If

$$\Delta < t_N (\mu_h - \mu_l) [\bar{p}_D^S - \bar{p}_D^L] + \mu_h$$

MA will choose the OM strategy but SA can render MA's use of obfuscation ineffective if the optimal values of  $p_D^S$  and  $p_D^L$  satisfy

$$p_D^S - p_D^L = \frac{\Delta}{t_N (\mu_h - \mu_l)} \quad (6)$$

and

$$(1 - t_N) (p_D^S)^{\frac{1-r}{r}} + t_N (p_D^L)^{\frac{1-r}{r}} = \frac{r\lambda(d - c)}{(1 - \lambda)c} \quad (7)$$

Condition (6) is condition (3) in equilibrium. (7) can be obtained by replacing  $p_D^S$  and  $p_D^L$  from (4) and (5), respectively, in the left hand side and simplifying.

From (3) (4), (5), (6) and (7) we obtain the following propositions.

*Proposition 1:* For a given cost of obfuscation (fixed  $\Delta$ ),  $p_D^S - p_D^L$  is increasing in  $t_M$  and  $t_N$  if

$$\Delta \geq t_N (\mu_h - \mu_l) [\bar{p}_D^S - \bar{p}_D^L] + \mu_h$$

and is decreasing in  $t_M$  and  $t_N$  otherwise.

If developing obfuscated malware is relatively costly (high  $\Delta$ ), MA would not choose to develop obfuscated malware, and hence, as the selector's accuracy improves, SA designs more lenient classifier for those files selected as normal and a more stringent classifier for those files selected as malware. Thus, as conventional wisdom would suggest, the differentiation between the detection rates for the two types of files increases when selector becomes better at discriminating the two types. However, if the cost of obfuscation is sufficiently low, MA would use obfuscation to beat the selector. But SA can render MA's use of obfuscation ineffective by making the classifier for files selected as malware less stringent and the classifier for files selected as normal more stringent with an increase in selector's accuracy.

*Proposition 2:* For a given selection accuracy (i.e., fixed  $t_M$  and  $t_N$ ),  $p_D^S - p_D^L$  is increasing in the cost of obfuscation as long as

$$\Delta < t_N (\mu_h - \mu_l) [\bar{p}_D^S - \bar{p}_D^L] + \mu_h$$

and is constant otherwise.

When, for example, the detection rate of both classifiers is same, i.e., their difference is 0, MA will not invest on obfuscation as gaming the selector is of no use. On the other hand, when the detection rate of one classifier is 1 and the other is 0, i.e., the difference is maximum, MA will try her best to game the selector by putting more effort for obfuscation such that malware is directed to the classifier with the detection rate 0. Of course, this configuration may not give optimal expected payoff for SA due to increased false positives.

*Proposition 3:* For a given detection rate of the classifiers (fixed  $p_D^S - p_D^L$ ), the cost of obfuscation required to render MA's use of obfuscation ineffective, increases with the selector's accuracy.

This proposition implies that with the increase of selector's accuracy, MA has more incentive to develop obfuscated malware. Therefore, to fully realize the benefits of having a selector to pre-screen incoming files, it should be costly enough for MA to develop obfuscated malware.

## VI. CONCLUSION

In this paper, we construct a stylized game theoretic model to analyze the optimal configuration of a heterogeneous AV System with a selector component that pre-screens incoming files. Our model incorporates one crucial aspect of the game:

the strategic behaviour of malware authors who would invest to develop obfuscation techniques trying to beat the selector component of the AV System. Based on the analysis of our model, we obtain the following implications for the design and configuration of the classifiers.

When the cost of obfuscation for a malware author is sufficiently low, the difference between the optimal detection rates configured for the two classifiers will be decreasing in the selector's accuracy. This implies that when the selector's accuracy increases, it is optimal for a security analyst to maintain a less stringent classifier for malware and a more stringent classifier for normal files. Thus, SA can render MA's use of obfuscation ineffective by decreasing the difference of the detection rates of the two classifiers. Also, it was shown that the minimal cost for developing obfuscated malware sufficient to game the selector increases with the accuracy of the selector. Therefore, the cost of obfuscation should be considered an important factor, in addition to the discriminatory power of detection, when designing an AV System.

#### REFERENCES

- [1] P. Szor, *The Art of Computer Virus Research and Defense*. Addison-Wesley, 2005.
- [2] C. Nachenberg, "Computer virus-antivirus coevolution," *Communications of the ACM*, vol. 40, no. 1, pp. 46–51, January 1997.
- [3] S. Abramsky and R. Jagadeesan, "Games and full completeness for multiplicative linear logic," *Journal of Symbolic Logic*, vol. 59, pp. 543–574, 1994.
- [4] N. Dalvi, P. Domingos, Mausam, S. Sangha, and D. Verma, "Adversarial classification," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 99–108.
- [5] A. Jafari, A. R. Greenwald, D. Gondek, and G. Ercal, "On no-regret learning, fictitious play, and nash equilibrium," in *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 226–233.
- [6] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.
- [7] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.
- [8] A. Walenstein, R. Mathur, M. R. Chouchane, and A. Lakhota, "Normalizing metamorphic malware using term rewriting," in *Proceedings of the Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2006)*, 2006, pp. 75–84.
- [9] M. R. Chouchane, A. Walenstein, and A. Lakhota, "Statistical signatures for fast filtering of instruction-substituting metamorphic malware," in *Proceedings of the 2007 ACM Workshop on Recurring Malcode (WORM 2007)*, 2007, pp. 31–37.
- [10] J. P. Egan, *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.