Surviving in Cyberspace: A Game Theoretic Approach

Charles A. Kamhoua, Kevin A. Kwiat
Air Force Research Laboratory, Rome, NY 13441, USA
charles.kamhoua.ctr @ rl.af.mil, kevin.kwiat @ rl.af.mil

Joon S. Park
Syracuse University, Syracuse, NY 13244, USA
jspark @ syr.edu

Abstract—**As information systems become ever more complex and the interdependence of these systems increases, a mission-critical system should have the fight-through ability to sustain damage yet survive with mission assurance in cyberspace. To satisfy this requirement, in this paper we propose a game theoretic approach to binary voting with a weighted majority to aggregate observations among replicated nodes. Nodes are of two types: they either vote truthfully or are malicious and thus lie. Voting is strategically performed based on a node's belief about the percentage of compromised nodes in the system. Voting is cast as a stage game model that is a Bayesian Zero-sum game. In the resulting Bayesian Nash equilibrium, if more than a critical proportion of nodes are compromised, their collective decision is only 50% reliable; therefore, no information is obtained from voting. We overcome this by formalizing a repeated game model that guarantees a highly reliable decision process even though nearly all nodes are compromised. A survival analysis is performed to derive the total time of mission survival for both a one-shot game and the repeated game. Mathematical proofs and simulations support our model.**

Index Terms— **Bayesian game, binary voting, cyberspace, fault-tolerant networks, fight-through, network security, survivability**

I. INTRODUCTION

The need for survivability is most pressing for mission-critical systems. As information systems become ever more complex and the interdependence of these systems increases, the survivability picture becomes more and more complicated. Unfortunately, it is not always possible to anticipate every type of component failure and cyber attack within large information systems and attempting to predict and protect against every conceivable failure and attack soon becomes exceedingly cumbersome and costly. Additionally, some damage results from novel, well-orchestrated, malicious attacks that are simply beyond the abilities of most system developers to predict. Under these conditions, even correctly implemented systems cannot ensure that they will be safe from the possible damages caused by failures and cyber attacks. Therefore, a mission-critical system placed in cyberspace should have the fight-through ability to sustain damage yet survive with mission assurance.

Fault-tolerant networks have been an active research area for decades. However, the solutions proposed for malicious nodes are still not robust. Generally, faults can be classified in two types: unintelligent and intelligent. Unintelligent or benign faults are irrational and random. Intelligent faults can be classified into two categories: faults from selfish nodes that manifest as not following group protocol or not conforming to security protocols that fail to promote their self-interest; and malicious faults that manifest as a node seeking to inflict maximum damage to the system. Thus, it is evident that malicious faults poses the most danger to a system; yet, any system administrator must rationally oppose malicious nodes by minimizing the dangers they present. Game theory is the unifying mathematical framework that can model the rational conflict between a system's administrator and the malicious nodes, and can scrutinize the possible solutions with a precise characterization of their properties. This paper proposes a new game theoretic model to deal with malicious nodes. Game theory is one of the most promising approaches because the malicious attacks faced by system administrators are becoming exceptionally sophisticated and game theory deals with intelligent players.

The aim of this work is to inspect critical applications that require the monitoring of a binary event to make a binary decision. For more than two centuries, binary voting has attracted several researchers from different disciplines including political science, philosophy, mathematic, game theory and more recently computer science and engineering. In fact, research in binary voting started in 1785 by the French Mathematician and Philosopher, the Marquis de Condorcet. Condorcet's research currently finds tremendous applications in fault-tolerant networks. As a strong proponent of democracy, Condorcet proved the well-known Condorcet Jury

Manuscript received March 9, 2012; accepted March 27, 2012. Approved for Public Release; Distribution Unlimited: 88ABW-2011-6296 Dated 05 December 2011.

doi:10.4304/jcm.7.6.436-450

Theorem (CJT) [16]. The CJT shows that in a population in which individual opinions are better than random, the opinion of the majority is superior to that of a smart dictator or a small elite group. In the context of fault-tolerant networks, instead of letting a single node (a dictator) makes a critical decision, nodes are replicated and the final decision is generally the result of a voting mechanism in which the result is the majority's opinion. This mechanism serves two purposes in engineering design: it increases the decision accuracy, and it can tolerate the failure or compromise of a minimum number of nodes. These two purposes are critical for fighting through attacks on applications in which the enemy's strategy is to compromise some nodes.

It is worth mentioning that even though binary voting seems to be an old subject, philosophers and political scientists over the centuries have focused on truthful voters having similar interest but different private information to reach the correct decision. Engineers are thus faced with voters having a complicated mix of untamable faults that induce malicious behavior encompassing strategic attempts to defeat the voting so as to create an incorrect decision.

The main contribution of this research is to use new insight from game theory and mechanism design to mitigate malicious node behavior in binary decision. Any network defender or decision maker can take advantage of, or gain insights from, the research results in this paper. We formulate an original Bayesian zero-sum game to model the conflicting and rational confrontation between a system administrator and the malicious nodes' objectives. The system administrator's objective is to maximize the aggregate node decision reliability while the malicious nodes want to minimize that reliability. To adhere to game theory's attribute of player rationality, we dismiss the case of benign but untamable faults since failures caused by such faults would not exhibit rational behavior. However, any model that is able to examine random behavior can be combined with the present work. Thereafter, we challenge the recurrent assumption that the number of faulty nodes is fixed and develop a dynamic node compromising model. Node compromising is analyzed in real time. Our dynamic model shows the change in equilibrium behavior over time as the number of compromised nodes increases. We also design a robust mechanism based on a repeated game and voter reputation. Further, our repeated game mechanism achieves high decision reliability under the compromising of nearly all nodes. Our model is supported by mathematical proofs and simulations.

This work proposes a robust and detailed binary voting framework for fault-tolerant networks. The remainder of the paper is organized as follows. Section II is dedicated to the related works. Section III presents an overview of game theory. A reader familiar with game theoretic concepts may skip that section. Section IV exposes our Bayesian Zero-sum game with its resulting equilibrium. Section V is about the dynamic analysis of our game. Section VI reveals our repeated game model. Section VII exhibits our simulation results and Section VIII concludes the paper.

II. Related Works

We summarize in this section a few researches in binary voting. Kwiat *et al.* [1] analyzed the best way to aggregate the node observations given the nodes' reliability. The nodes are assumed to be homogeneous. The reliability of a single node *p* is its probability to make the correct decision. They showed that Majority Rule (MR) performs better if the node' observations are highly reliable (*p* close to 1). But for low value of *p*, $\left(p < \frac{1}{2}\right)$ choosing a Random Dictator (RD) is better than MR. Random Troika (RT) combines the advantage of those two strategies when the node reliability is unknown ($0 \leq p \leq 1$). Generally, it can be shown that if the minority of nodes are compromised and $0.5 < p \leq 1$, assuming that an odd number of nodes is used, we will have MR=Random N>….>Random 5>RT>RD. However, if the majority of nodes are compromised, the previous inequality is reversed. That is because, with a majority of compromised nodes, increasing the size of the subset of deciding nodes also increases the likelihood of compromised nodes taking part of the decision. The research in [2] proposes a witness-based approach for data fusion in wireless sensor networks.

Wang *et al.* [3] analyzed the nodes decision in a cluster. There are *n* clusters of *m* nodes, with a total of *n*m* nodes. The attacker chooses the number of clusters to attack while the defender chooses how many nodes participate in the decision in each cluster. The authors formulate a zero-sum game in which the defender maximizes the expected number of clusters deciding correctly while the attacker minimizes that number. They proposed a general framework to find the Nash equilibrium of such game. However, the cluster structure is assumed to be fixed. In fact, the defender has a better optimization strategy just by changing the cluster structure.

Park *et al.* [4-5] proposed a trusted software-component sharing architecture in order to support the survivability at runtime against internal failures and cyber attacks in mission critical systems. They defined the definition of survivability using state diagrams, developed static and dynamic survivability models, and introduced the framework of multiple-aspect software testing and software-component immunization.

Bhattacharjee *et al* [6] used a distributed binary voting model in cognitive radio. To compensate their noisy observation of channel utilization by primary spectrum users, each secondary user requests their neighbor's opinion (vote). Those interactions are repeated and the Beta distribution is used to formulate a trust metric. Nodes with low trust are eliminated to have a more accurate channel evaluation.

Ma and Krings [7] propose evolutionary game theory as a method to analyze reliability, survivability, and fault tolerance. They consider the agreement algorithm of the well-known Byzantine generals' problem in a dynamic

environment. In their formulation, the number of generals as well as the number of traitors may change over time.

Malki and Reiter [8] analyze Byzantine quorum systems. They propose a masking quorum system in which data are consistently replicated to survive an arbitrary failure of data repositories. Their work also proposes a disseminating quorum system. Faulty servers can fail to redistribute the data but cannot alter them.

Gao *et al*. [9] use Hidden Markov Model (HMM) to detect compromised replicas. Their results show greater detection accuracy compared to others. Other work applying game theory to network security can be found in [10-13]. Becker *et al.* [11] develop an attacker-defender game in virtual coordinate system. The defender's goal is to maximize the accuracy of the computed virtual coordinate while the attacker is opposed to that outcome. The defender implements a space and temporal outlier detection using a fixed or adaptive threshold to eliminate the malicious nodes' reports. In this paper, the primary goal is intrusion resilience as opposed to intrusion detection. Thus, malicious nodes' reports (vote) are not eliminated but they are constrained to adopt a strategy that eliminates their impact on the game outcome. Detailed surveys of game theory applied to network security can be found in [14-15].

Alongside the research above, there is a significant mathematical literature about binary voting starting with Condorcet [16]. Simply stated, the CJT shows that if a group of homogeneous and independent voters, with voter competence better than random, uses the simple majority rule to choose among two alternatives having equal *a priori* probability, then the group decision accuracy monotonically increases and converges to one as the number of voter increases. Owen *et al.* [17] generalized the CJT while considering any distribution of voter competence. The original CJT was restricted to a uniform distribution of voter competence or reliability *p*. The original CJT also considered that each player voted sincerely. Myerson [18] proved the CJT with strategic voters. In his model, the number of players is a random variable drawn from a Poisson distribution. Laslier and Weibull [19] investigate the general condition in which truthful voting is a Nash equilibrium. In their work, the players are equally competent but differ in their belief about the *a priori* likelihood of the two states of nature. Moreover, each player's reliability regarding the two states of nature are distinct. Players also differ in their valuation of the cost associated with the two types of error (type I and type II error). However, all players agree on what decision should be taken in each state. Shapley and Grofman [20] examine binary voting in which the voters' choices are statistically dependent. They show that nonmonotonic rules can eventually be superior to monotonic ones. For instance, in a committee of three members in which one corrupt member always votes the opposite way, a unanimous vote must be wrong and thus rejected. Goodin and Estlund [21] scrutinize the CJT when the voters' reliabilities are unknown. In their approach, there are two *ex post* interpretations when 80% of a large population agrees to one of two outcomes in a binary election. First, the majority opinion may be correct in which case the average voter's competence is 0.8. Second, the majority opinion may as well be incorrect and then the average voter's competence should be 0.2. A mathematical survey of binary voting is provided in [22].

III. Overview of Game Theory

According to Myerson [23], game theory can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. A rational player makes decisions to satisfy his or her self interest. A game can be represented in different forms. Most of the time, it is represented in either extensive form or in strategic (or normal) form. The extensive form of a game is used to formalize sequential action of players. In those games, the order in which players act in the game is important. On the other hand, a strategic form of a game is formalized as:

$$\Gamma = (N, (S_i)_{i\in N}, (u_i)_{i\in N}).$$

N is the set of players of game Γ, *i* is a player in *N*, S_i is the set of pure strategies that players *i* can choose from, u_i is the utility function of player *i*. A mixed strategy is a random combination of two or more pure strategies. A strategy profile is a combination of strategies that the players can choose. The set of all possible strategy profiles is $S = X_{j\in N} S_j$. u_i is a function defined from the set of strategy profiles *S* to the set of real numbers *R*. At any strategy profile, the utility function associates the expected utility payoff that player *i* would get. A game in strategic form can also be represented by a matrix as in Table I.

For two strategies *A* and *B*, *A* strictly dominates *B* if *A* always earns a higher payoff than *B*. *A* weakly dominates *B* if *A* never earns a lower payoff than *B* and *A* is superior to *B* for at least one strategy.

A. Nash Equilibrium

A strategy profile is a Nash equilibrium if and only if no player can gain by changing its strategy when other players do not change. Moreover, in a Nash equilibrium, each player's equilibrium strategy is a best-response to other players' equilibrium strategies [23].

Definition 1: A mixed strategy profile σ^* is a Nash equilibrium if, for all players *i*,

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \text{ for all } s_i \in S_i \quad (1)$$

We have a strict Nash equilibrium if inequality (1) is strict.

Theorem 1: Given any finite game Γ in strategic form, there exists at least one Nash equilibrium.

Theorem 2: A strictly dominated strategy is never a Nash equilibrium.

Theorem 2 comes from the fact that a strictly dominated strategy can never be optimal.

B. Bayesian Game

A Bayesian game, or game of incomplete information, is a game in which some payoff information is not common knowledge; instead it is private information [23]. A player's private information is captured by its type. In a Bayesian Nash equilibrium strategy profile, each player maximizes his expected utility given his type and his belief about the distribution of other players' types. For instance, compromised nodes do not have the same motivation as uncompromised nodes and then, they have different utilities. Moreover, an uncompromised node cannot distinguish between uncompromised and compromised nodes and must play its optimum strategy according to its belief about the number of compromised nodes.

IV. Proposed Game Theoretic Analysis

This section provides an analysis of strategic voting. We also provide a general equilibrium property of binary voting game with malicious voters. A node may be a sensor or radar involved in any monitoring activity. For example, radars are used in a battlefield to monitor enemy activity in a given region. In respect to wireless sensor networks, monitored data are transmitted via multi-hop communication. This work does not consider multi-hop interaction in the presence of malicious voters; instead, the monitored results (votes) are directly transmitted in one hop communication (via wired or wireless connection) to a trusted decision center in charge to aggregate the votes. Given the criticality of the decision to be made, a single node cannot be trusted. Rather, several nodes monitor the same environment or event. The number of nodes is generally odd to avoid a tie when using the simple majority rule.

Another problem to consider is the possibility of a malicious node to abstain. However, abstention has two drawbacks for malicious nodes: it facilitates their detection at the decision center, and abstention is a dominated strategy. A strategic vote from a malicious node (either true or false) is always better than abstention. Therefore, this work assumes that there is no abstention.

Without loss of generality, we assume in this paper that the nodes are homogeneous and that each node's reliability is p. We also consider that $0.5 < p \leq 1$. Therefore, in the framework of Condorcet [16], using a simple majority and without malicious nodes, the certainty of the decision monotonically increases and converges to one as the number of voters grows to infinity.

A. Game Model

The players are the nodes. The set of players is $N = \{1,2,\dots n\}$. There are two types of nodes: regular (r) or compromised (c). The set of types is $\Phi = \{r, c\}$. Let q be the probability distribution over the type of players. Specifically, we have:

$$\begin{cases} q(c) = \lambda, \text{with } 0 \leq \lambda \leq 1, \text{and} \\ q(r) = 1 - \lambda \end{cases} \quad (2)$$

The proportion of compromised nodes is respresented by λ. We consider λ as well as the total number of nodes to be common knowledge among the nodes. We will relax this assumption in Section V where we conduct a dynamic analysis.

There are two states of the nature with equal *a priori* probability. The state-of-nature is a set $\Theta = \{s, a\}$, whose element s indicates that the monitored target is safe and the element a indicates that the target is under attack.

At a given time, the nodes are required to observe the state of nature and cast a binary vote indicating their observation of that state. Each node has two available strategies: report its observation truthfully (T) or falsify its observation (F). Thus, the set of attackers' strategies is $\Psi = \{T, F\}$. Likewise, the defender (decision maker or network owner) has a binary choice. Considering that there are compromised nodes, the defender chooses between the majority's opinions (M) or the minority's opinions (m) to detect the correct state of nature. We will see that choosing the minority's opinions may be more effective when the majority of nodes are compromised. Thus, the set of defender's strategies is $Z = \{M, m\}$.

All *regular* nodes are under the control of the defender and they always vote truthfully. Without loss of generality, we distinguish three cases for malicious nodes. First, all the compromised nodes are under the control of a single attacker. Second, there are several attackers but all of the attackers collude to mislead the voting mechanism. Third, there are multiple independent attackers adopting identical strategy, an attacker symmetric strategy profile. In all three cases, the malicious nodes have the same motivation and can mathematically be modeled as a single player: the attacker. We use this approach to model the voting game as a two-player's game: an attacker-defender game.

Considering the case of independent malicious nodes, adopting a different strategy will necessitate at least as many players (or as many independent games) as there are malicious nodes. This is the approach that will be adopted in Section VI.

Moreover, since the attacker and the defender have strictly conflicting objectives, our voting game will be modeled as a two-player zero sum game. The utility function or payoff of the defender will be the system-of-nodes' reliability or the probability to find the correct state of nature while that of the attacker is the opposite. The game also considers the following practical situations. First, the attacker distinguishes between the compromised and the regular nodes while the defender does not. Otherwise, the defender should simply remove the compromised node from the system to maximize its utility. Therefore, our game model is a Bayesian game. Second, we assume that nodes have common knowledge of the game; each node is rational and wants to maximize its expected utility. Therefore, the case of faulty unintelligent nodes with random irrational behavior is outside the scope of the current paper. Finally, when a regular node truthfully reports its observation, its reliability or its probability to make a correct observation is p and that of an incorrect observation is $1-p$. Recall our assumption that $0.5 < p \leq 1$. In fact, a fair coin tosses

will perform better if we have $0 \leq p \leq 0.5$. Recall that regular nodes always vote truthfully.

Let A be the probability that the majority's opinion is correct when the malicious nodes vote truthfully. Similarly, let B be the probability that the majority's opinion is correct when the compromised nodes falsify their observation. It can be seen that A and B are the system-of-nodes' reliability when the attacker uses the strategies (T and F respectively) and the defender uses the majority's opinion. Further, the defender has a binary choice: the majority or minority opinion. Therefore, the probability that the minority's opinion is correct when the malicious nodes play T (F respectively) must be *1-A* (*1-B* respectively). Also, the game is zero-sum. Then, the attacker's payoffs are calculated by taking the opposite of the defender's payoffs. We summarize those payoffs in the strategic form of the game as represented in Table I.

TABLE I: VOTING GAME IN NORMAL FORM

		Defender	
		M	*m*
Attacker	*T*	*-A; A*	*A-1; 1-A*
	F	*-B; B*	*B-1; 1-B*

B. Equilibrium Analysis

This subsection explores all possible Nash equilibrium profiles in the voting game in Table I. From the payoff definition above, we must always have $A \geq B$. This is because we assume that $0.5 < p \leq 1$ and then the majority's opinion is more reliable when malicious nodes play T as opposed to F. In addition, since increasing the number of nodes increases the majority's opinion reliability and a single node's reliability is $p > 0.5$, we must have $A \geq p > 0.5$.

The value of the payoff B is more ambiguous. However, it is evident that B monotonically decreases with the proportion of compromised nodes λ. We note that function $B(\lambda)$. To be specific, if there is no compromised node, all nodes are truthful and we have $B(\lambda) = B(0) = A > 0.5$. In the other extreme case, if all the nodes are malicious and play F, a malicious node may reveal the truth state, but with a probability less than 0.5 because $0.5 < p \leq 1$, then the majority's opinion is more likely to be wrong. This means that $B(\lambda) = B(1) < 0.5$. Thus, when we consider the discrete nature of the number of malicious nodes and λ, by the intermediate value theorem, there must be a number λ_0 $(0 < \lambda_0 < 1)$ such that,

$$\begin{cases} B(\lambda) > 0.5 \text{ if } \lambda < \lambda_0 \text{ and} \\ B(\lambda) \leq 0.5 \text{ if } \lambda \geq \lambda_0 \end{cases} \quad (3)$$

In summary, when $\lambda < \lambda_0$ we have

$$A \geq B > 0.5 > 1 - B \geq 1 - A. \quad (4)$$

The defender's dominant strategy in Table I is M. The defender chooses the majority's opinion. The attacker's best response is to play F. Thus, (F, M) is the only Nash equilibrium profile.

When $\lambda \geq \lambda_0$ we have

$$A \geq 1 - B > 0.5 > B \geq 1 - A. \quad (5)$$

There is no dominant strategy. If the attacker believes that the defender will choose M, its best response is to play F. If the attacker plays F, the defender' best response is the minority's opinion or m. However, if the attacker believes that the defender will choose the minority's opinion, he is better off telling the truth or playing T. If the attacker plays T, the defender will prefer M and so on. Clearly, there is no stable solution. No pure strategy Nash equilibrium exists. As a fact, any game in strategic form has a Nash equilibrium. Thus, there must be a mixed strategy Nash equilibrium.

Let $\tau = \alpha T + (1 - \alpha)F$ be the mixed strategy Nash equilibrium of the attacker. From basic principles of game theory, the attacker optimum strategy is to randomize or choose α such that the defender is indifferent when choosing between the strategy M and m. This means that we must have for the defender:

$$\begin{aligned} U_D(\tau, M) &= U_D(\tau, m) \\ \Rightarrow \alpha A + (1 - \alpha)B &= \alpha(1 - A) + (1 - \alpha)(1 - B) \\ \Rightarrow \alpha &= \frac{1 - 2B}{2(A - B)}. \end{aligned} \quad (6)$$

Then,

$$U_D(\tau, M) = U_D(\tau, m) = \alpha A + (1 - \alpha)B = 0.5 \quad (7)$$

Similarly, let $\sigma = \beta M + (1 - \beta)m$ be the mixed strategy Nash equilibrium of the defender. The defender randomizes between T and F in such a way that the attacker is indifferent between voting truthfully and falsifying the vote. This can be translated by:

$$\begin{aligned} U_A(T, \sigma) &= U_A(F, \sigma) \\ \Rightarrow \beta(-A) + (1 - \beta)(A - 1) &= \beta(-B) + (1 - \beta)(B - 1) \\ \Rightarrow \beta &= 0.5 \end{aligned} \quad (8)$$

Then,

$$\begin{aligned} U_A(T, \sigma) &= U_A(F, \sigma) \\ &= 0.5(-A) + 0.5(A - 1) = -0.5. \end{aligned} \quad (9)$$

Consequently, at the mixed Nash equilibrium profile $(\tau, \sigma) = (\alpha T + (1 - \alpha)F; 0.5M + 0.5m)$, the defender payoff or probability of a correct aggregate observation is 0.5 and that of the defender is the opposite.

From (6), the value of α is positive only if $B < 0.5$, which only happens when the attacker has compromised more than the proportion of nodes λ_0. One interpretation of α is to be the probability by which each compromised node votes truthfully. Another interpretation is that the attackers organize the compromised nodes such that a proportion α votes truthfully while the proportion $(1 - \alpha)$ falsifies their vote. In this case, if the voting game is repeated and a simple majority rule is used, a proportion α of compromised nodes may exhibit the same voting record as regular nodes and there will be no easy way to detect them. Observe also that a single attacker

recommending all the malicious nodes to play *T* with probability α has a quite similar effect than numerous independent attackers that play *T* with probability α. Thus, modeling all malicious nodes as a single attacker and restricting our study to a two-player's game does not change the game's dynamic that much.

It is important to note that the game we have designed in this section and the Nash equilibrium analysis we perform are general enough to capture most of the fundamental essences of binary voting games with malicious voters. We did not assume any specific number of nodes. This analysis shows that the minimum decision reliability or defender's payoff is 0.5 even though all the nodes are malicious. Further, if more than a critical proportion of nodes λ_0 are compromised, the defender cannot improve the system of nodes' reliability above 0.5. In this case, the nodes' binary votes give no information to the defender. One may believe that when more than a critical proportion of nodes λ_0 are compromised, a defender should just choose the minority's opinion and achieve a payoff above 0.5. However, this belief underestimates the attacker's intelligence and his ability to understand the rule of the voting game. We highly discourage such belief as it increases the system vulnerability.

C. Attack Strategies Classification

In a voting game, malicious nodes can perform several types of strategic attacks. These attacks can be divided in two general categories: symmetric and asymmetric attack. In a symmetric attack, compromised nodes not only coordinate their actions but also communicate to perform a partial aggregation of their results with the goal of reporting the same observation. After observing the state of nature, the compromised nodes first communicate with each other. They perform an exploratory vote among themselves and find their majorities' opinion. That opinion will be more reliable as the number of compromised nodes grows. The compromised nodes' majority opinion is reported by the entire set of compromised nodes if they choose the strategy *T* and the opposite is reported when they choose *F*. In this case, compromised nodes behave as a block. All the λN compromised nodes play *T* or they play *F*.

A symmetric attack is also possible when the attacker has other means to detect the state of nature besides monitoring. This is the case when the action being monitored is performed by the attacker itself. For instance, if an airplane belonging to the attacker is under radar monitoring and the attacker also controls the malicious nodes, then all the malicious nodes should know the state of nature with certainty. Thus, in a symmetric attack, we consider that when a node falsifies its observation, its probability to make an incorrect vote is one and that of a correct vote is zero.

On the other hand, in an asymmetric attack, compromised nodes coordinate to mislead the system but do not have time to communicate or have an exploratory vote. Moreover, malicious nodes do not have any other way to detect the state of nature besides monitoring. Actually, a monitored airplane may belong to a different enemy than the attacker controlling the malicious nodes. In this case, malicious nodes can falsify their result but do not necessarily report the same results. The reported results depend on each node's individual observation. Therefore, when the attacker chooses to play *F*, each node falsifies its observation. However, the probability that a node reports the correct state of nature after vote falsification is *1-p*.

This is justified by the fact that when a node observes, for instance, that the target is under attack but chooses to falsify its observation and reports that the target is safe, there is still a probability *1-p* that the target is really safe. That is because *1-p* is the probability of an incorrect observation. Similarly, p is the probability of a correct observation or a correct vote when a malicious node strategically chooses to report truthfully under asymmetric attack.

Another way to understand the main difference between an asymmetric attack and a symmetric attack is to use the average node decision reliability $\bar{p}$. An increase in the proportion of malicious nodes λ causes the decrease of $\bar{p}$. In case of an asymmetric attack, we have:

$$\bar{p} = (1-\lambda)p + \lambda(1-p) \tag{10}$$

This is because malicious nodes still tell the truth with probability $(1-p)$. We can verify in (10) that, if $0.5 < p \leq 1$ as we assume, we will have $\bar{p} > 0.5$ when $\lambda < 0.5$ and $\bar{p} < 0.5$ when $\lambda > 0.5$.

However, in case of a symmetric attack, malicious nodes never tell the truth. Thus, we have:

$$\bar{p} = (1-\lambda)p + \lambda * 0 = (1-\lambda)p \tag{11}$$

Similarly, if we take $0.5 < p \leq 1$, we will have $\bar{p} > 0.5$ when $(1-\lambda) > \frac{1}{2p}$ and $\bar{p} < 0.5$ when $(1-\lambda) < \frac{1}{2p}$.

To summarize, if the number of nodes is large, the defender should use the majority's opinion if $\bar{p} > 0.5$ and otherwise use the mixed strategy Nash equilibrium described above. The Nash equilibrium of the game can also be formulated as a function of the average node reliability. Specifically, for a large number of nodes, (3) can be reformulated as:

$$\begin{cases} \lambda < \lambda_0 \Rightarrow B(\lambda) > 0.5 \Leftrightarrow \bar{p} > 0.5 \text{ and} \\ \lambda \geq \lambda_0 \Rightarrow B(\lambda) \leq 0.5 \Leftrightarrow \bar{p} \leq 0.5 \end{cases} \tag{12}$$

The critical proportion of compromised nodes λ_0 can simply be interpreted as the minimum proportion that will make the average node reliability $\bar{p}$ falls below 0.5. However, this is generally not true for small number of nodes. The work in [20] provides a few contradictory examples with 3 to 5 nodes in which $\bar{p} < 0.5$ and $B > 0.5$ and vice versa.

V. DYNAMIC ANALYSIS

In Section IV, the ratio of compromised nodes λ was common knowledge among the players. However, in practice, the defender can only estimate the number of compromised nodes and hereby λ. This section considers a single attacker with compromised nodes that know the true state of nature. The compromised node may know

the true state of nature if the attacker is also responsible of the event being monitored. The attacker knows the exact number of compromised nodes while the defender can only estimate that number. To estimate the number of compromised nodes, the defender follows the framework described in [24-25].

The framework in [24-25] divides the attacker's action into three statistical processes:

Process 1-when the attacker has identified one or more known vulnerabilities and has one or more exploits on hand.

Process 2-when the attacker has identified one or more known vulnerabilities but doesn't have an exploit on hand.

Process 3-when no known vulnerabilities or exploits are available.

Initially, we consider the nodes to have no known vulnerabilities and no known exploits are available to the attacker. This assumption is reasonable for critical applications in which no node should be deployed with a known vulnerability. Therefore, in the estimation of the mean time to compromise a node, process 1 and process 2 [24-25] are eliminated and we are only left with process 3. Process 3 is the identification of new vulnerabilities and exploits. Researchers in [24-26] suggest that the vulnerability-discovery rate is constant over time. We also take the pessimistic assumption that the attacker has the expertise to exploit a new vulnerability as soon as they are discovered. Therefore, the time to compromise a node will have an exponential distribution. Clearly, for a given node at time *t*, the probability that that node is compromised is

$$F(t) = 1 - e^{-rt}, r, t > 0. \quad (13)$$

We consider the *N* nodes that are independently designed and fabricated to increase their security and fault tolerance. In fact, diversity is one of the fundamental characteristics of fault-tolerance. Providing multiple different design fabrications and implementations denies the attacker the power to exploit the same vulnerability to compromise all the replicas. The correlation between the nodes failure is eliminated. Thus, the node failures are assumed to be statistically independent and the probability distribution of the number of compromised nodes *l* is a binomial distribution or,

$$\psi_l(t) = \binom{N}{l}(1 - e^{-rt})^l (e^{-rt})^{N-l}. \quad (14)$$

The expected number of compromised nodes is:

$$E_l[\psi_l(t)] = N(1 - e^{-rt}). \quad (15)$$

Moreover,

$$\lim_{t\to 0} \psi_0(t) = 1 \text{ and } \lim_{t\to +\infty} \psi_N(t) = 1 \quad (16)$$

Therefore, the number of compromised nodes is initially zero and increases with time up to *N*. The fixed proportion of compromised node λ is replaced by $\lambda(t) = 1 - e^{-rt}$.

Let $P(k|l)$ be the probability that *k* nodes report the correct state of nature given that *l* nodes are compromised and play the strategy *F*. We consider an odd number of nodes. Let $m = \frac{N+1}{2}$. The law of total probability indicates that the simple majority rule will be correct with probability

$$B(t) = \sum_{l=0}^{N} \sum_{k=m}^{N} P(k|l)\, \psi_l(t) = \sum_{l=0}^{N} \psi_l(t) \sum_{k=m}^{N} P(k|l)$$
$$= \sum_{l=0}^{N} \binom{N}{l}(1 - e^{-rt})^l (e^{-rt})^{N-l} \sum_{k=m}^{N} P(k|l). \quad (17)$$

Recall that *B* is the system-of-nodes' reliability when the attacker uses the strategy *F* and the defender uses the majority's opinion. In this dynamic analysis, *B* is obviously a function of time and therefore, represented as $B(t)$.

A. Average Node Reliability and Time of Mission Survival

The reliability of the system of nodes can be captured by the average node reliability. We consider the pessimistic case when the attacker is responsible of the even being monitored. Then all the malicious nodes know the true state of nature. Thus, when the malicious nodes play *F*, the average node reliability is:

$$\bar{p} = \frac{pNe^{-rt} + 0N(1 - e^{-rt})}{N} = pe^{-rt} \quad (18)$$

The mission can survive or keep minimum acceptable level of functionality as long as $B(t) > 0.5$. For a large number of nodes, this is equivalent to $\bar{p} > 0.5$. Then,

$$\bar{p} > 0.5 \Rightarrow pe^{-rt} > 0.5 \Rightarrow t < \frac{\ln 2p}{r}. \quad (19)$$

Equation (19) shows the maximum mission time. It allows us to avoid solving the more complex equation in (17).

We can see that $\bar{p}(t)$ decreases over time. The critical moment at which the defender starts to randomize as described in Subsection IV-B is when $B(t) = 0.5$ or $\bar{p} = 0.5$.

Obviously, the time of mission survival (19) can be extended if the nodes are repaired during the mission. The mission can survive even longer if the rate of node repair is higher than the rate of node compromising. However, in a contested cyber environment, the severity of threats (including completely unforeseen attacks – known as zero-day attacks) and intensity of attacks can preclude the use of repair to assure a mission. Repair of a compromised node can be a time-consuming process involving a forensics analysis and the dispatching of computer-recovery personnel. Without adequate defenses, attacks perpetuated during the repair period can deplete any spare on-line nodes and thus cause mission failure. We propose in the next section a repeated game framework coupled with a reputation mechanism to overcome this shortcoming.

VI. REPEATED GAME ANALYSIS

Section V shows that for the one-shot voting game the result heavily depends on when the vote takes place. Above a given time, the system reliability is 50%. In the context of binary vote, this means that the vote gives no information to the defender. Moreover, the maximum mission time we calculated appears to be short (19). For instance, if we consider that the nodes are 80% reliable ($p = 0.8$), the mean time to compromise a node is 30 days and is exponentially distributed $\left(r = \frac{1}{30}\right)$, then the mission survival time must only be 14 days as calculated using (19). We are now presenting a repeated game model reinforced by rules that substantially increase that survival time. In fact, fault-tolerance is not just the property of individual machines; it may also characterize the rules by which they strategically interact over time.

In Section VI-A, the defender formulates optimum rules governing multiple independent and self-interested nodes (regular and malicious), each with private information about their preferences. Once the rules are in place, the nodes play their optimum strategy in a repeated game in Section VI-B. The rules of the repeated game will be such that the nodes unavoidably achieve the defender's goal: squeeze potential attackers or malicious nodes into behaviors that are fault tolerable and extend the maximum mission survival time.

A. Repeated Game Rules Implemented by the Defender.

The defender periodically requests all nodes to cast a binary vote to report the state of nature. The initial reputation of all nodes is 0.5. We set the value of initial reputation as 0.5, considering that that reputation will increase for any node that perform better than random and decrease otherwise. Behaviors of nodes are used to build their future reputation. To avoid the possibility that malicious nodes may misbehave with impunity after accumulating a high reputation, the defender uses an exponentially weighted moving average to update the node's reputation. Specifically, a node *i* reputation $R_i(t)$ at times *t* is updated according to the following recursive formula.

$$\begin{cases} R_i(0) = 0.5 \\ R_i(t) = (1-\gamma)R_i(t-1) + \gamma \text{ if node } i \text{ vote correctly} \\ R_i(t) = (1-\gamma)R_i(t-1) \text{ if node } i \text{ vote incorrectly} \end{cases} \quad (20)$$

γ is the smoothing factor, $0 < \gamma < 1$.

Observe that the defender should choose a smoothing factor γ that is neither too large nor very small. A very large smoothing factor disproportionally allocates more weight to the last action. On the other hand, a too small smoothing factor will slow the increase of the reputation of regular nodes. A tradeoff value of the smoothing factor may be chosen around 0.1 to balance those opposing factors.

The research in [20] shows that in a heterogeneous group, with each voter characterized by its reliability p_i, the decision procedure that maximizes the likelihood of the aggregate decision to be correct is a weighted voting rule that assigns weights w_i such that:

$$w_i = log\frac{p_i}{1-p_i}. \quad (21)$$

The decision procedure of [20] only considers one-step voting with no malicious voter. We extend it by considering malicious nodes and repeated interactions. Moreover, voter competence p_i is replaced by their reputation $R_i(t)$ updated according to (20). In fact, in repeated voting interaction, a voter competence p_i can be derived from the frequency by which it votes correctly and that is captured in our reputation model. Clearly,

$$w_i = log\frac{R_i(t)}{1-R_i(t)}. \quad (22)$$

The vote weight in (22) takes advantage of misleading information from malicious nodes. For instance, if the defender knows that a specific malicious node lies all the time (e.g. the node has zero reputation, negative infinite weight). The information from that node should be inverted and used to get the true state all the time (100% sure). Equation (22) generalizes this concept when aggregating the vote from several nodes. Clearly, a node having a negative weight (or a reputation less than 0.5) has its binary vote flipped before computing the final result. Nodes with positive weight have their vote unchanged. A node with zero weight has its vote practically eliminated if at least one other node has a weight different to zero. We summarize our voting algorithm in Table II.

TABLE II: VOTING ALGORITHM

Repeated Binary Voting Algorithm
Initial Voting Round: $R_i = 0.5, w_i = 0$ for all nodes *i*.
If a node *i* vote is consistent with the majority of nodes,
Then, increase R_i according to (20).
Else, decrease R_i according to (20).
End If.
Update w_i according to (22).
Subsequent Voting Round:
//Nodes have different R_i and w_i from past votes
If a node *i* has a weight $w_i \geq 0$, keep node *i* vote unchanged.
Else, ($w_i < 0$) flip node *i*' s vote and changes the sign of w_i
Then, take the aggregate result as that of the outcome for which the nodes sum up more than the majority of weight.
End If.
Update R_i according to (20) and w_i according to (22) while taking the aggregate result to be the correct vote.

Initially, all nodes have a reputation of 0.5. Therefore, from (22), that corresponds to a weight of zero, since all nodes have the same weight, the weighting voting rule is similar to the simple majority rule. However, it is clear from the voting procedure of Table II that a node with zero weight has a vote that does not count if at least one other node has a weight different to zero.

As opposed to Section IV where the entire set of malicious nodes is mathematically modeled as a single

attacker, this section considers that each malicious node acts independently. Therefore, each malicious node is involved in an independent game against the defender.

Finally, the defender keeps the nodes' reputation private. This is to prevent a malicious node from using other nodes' reputation to mount an attack. Nevertheless, each node can infer its own reputation from its past behavior. Those inferences should be easier if the defender can record the node reputation without noise. However, since the defender takes the aggregate result as that of the outcome for which the nodes sum up more than the majority of weight and because that outcome does not perfectly detect the true state of nature, then each malicious node is involved in a game of imperfect monitoring with the defender. A game is of imperfect monitoring if the players cannot observe each other's actions without error. For instance, a node may observe the true state and vote truthfully while the aggregate result is incorrect because some nodes with high reputation have mistakenly observed the wrong state. That node will have its reputation decrease although it has voted correctly. In short, there are several reasons of inconsistencies between the defender's observation and a node's action. The next subsection analyzes a malicious node's optimum behavior under perfect monitoring as a starting point. We start by assuming perfect monitoring to facilitate the explosion. Subsection VI-C will deal with imperfect monitoring.

B. Malicious Nodes' Optimum Strategy in the Game

We consider that future node's payoffs are discounted by a factor δ $(0 < \delta < 1)$and that each malicious node wants to maximize its δ-discounted payoff average. Nodes are dynamically compromised according to the process we described in Section V. At the beginning of the game, there are only regular nodes; the rule of the game is common knowledge among all the nodes. We assume that any attacker that compromises a node will also acquire the rule of the game. Moreover, the attacker will believe that those rules are implemented because they are optimum to the defender as we will see in this section. Therefore, after compromising, each malicious node will play its optimum strategy given the voting rule.

Recall that regular nodes always vote truthfully and $0.5 < p \leq 1$. Then, after the first round, the reputation of regular nodes is more likely to increase above 0.5, and yields a positive weight (22) and a positive contribution into the aggregate decision. After all calculations, in the long run, the reputation of regular nodes $R_r(t)$ will oscillate around p and as a consequence, their weight will be close to

$$w_r = log\frac{p}{1-p} > 0 \tag{23}$$

w_r is the weight of a regular node.

Let us analyze the optimum decision of a malicious node that is only concerned with the short term benefit. Starting with a reputation of 0.5, if a malicious node falsifies the vote or play *F*, its reputation is more likely to decrease below 0.5, and yields a negative weight (22). A negative weight technically means that the defender will flip the malicious node's vote in the next round. Thus, a dishonest malicious node with a negative weight will see its vote have a positive contribution in the aggregate decision. Then a malicious node with a negative weight is better off being honest when it expects its vote to be flipped. Further, voting truthfully is also not optimum for a malicious node that has a positive weight. Therefore, a malicious node with a positive weight should play *F* while a malicious node with a negative weight must play *T*. We summarize a malicious node's vote consequence on the aggregate decision in Table III.

TABLE III: CONSEQUENCES OF A MALICIOUS NODE'S VOTE

Malicious node's weight	Malicious node's strategy	Consequences to the defender
Positive	Truthful (*T*)	Positive (*P*)
Positive	Falsify (*F*)	Negative (*N*)
Zero	Truthful (*T*)	Not any
Zero	Falsify (*F*)	Not any
Negative	Truthful (*T*)	Negative (*N*)
Negative	Falsify (*F*)	Positive (*P*)

Moreover, if a node weight is zero and at least one other node has a weight different to zero, the vote of a node with weight zero will simply not count. In fact, from (20), after the initial period, at least one node will have a weight different to zero. We take that into account in this analysis. Thus, a node that has a weight zero is simply indifferent between playing *T* or *F* and is willing to randomize between the two. It can be seen that, in the long run, the weight of a compromised node will be near:

$$w_c = log\frac{0.5}{1-0.5} = log(1) = 0 \tag{24}$$

To summarize, anytime a malicious node with a short term goal has a positive weight (reputation above 0.5), it should falsify its votes (play *F*). As a consequence, its reputation will decrease (20) and eventually fall below 0.5 to yield a negative weight (22). When a malicious node has a negative weight, it should vote truthfully (play *T*) to impose a negative consequence to the defender (see Table III). Equation (25) represents the malicious node strategy we just described.

$$\begin{cases} \text{Play } F \text{ if } w_i(t) > 0 \\ \text{Play } T \text{ or } F \text{ indiferently if } w_i(t) = 0 \\ \text{Play } T \text{ if } w_i(t) < 0 \end{cases} \tag{25}$$

Note that (25) is a deterministic strategy. We call this strategy *N* since it has a negative consequence to the defender as shown in Table III. Also, we can see from Table III that a deviation from the strategy (25) has a positive consequence to the defender and we will call that strategy *P*.

To continue our equilibrium analysis, note that w_i is a symmetric function of $R_i(t)$ around 0.5 (22). One consequence is that a malicious node with a reputation of 0.99 that chooses to play *F* has the same effect as a malicious node with a reputation of 0.01 that chooses to play *T*. This argument holds for any two reputation values that are symmetric around 0.5, say also 0.6 and 0.4.

Furthermore, looking into the long term effect, a malicious node with a positive weight that plays *T* loses the opportunity to damage the system in the current period but increases its reputation and weight for causing future damage. By the same token, a malicious node with a negative weight that plays *F* loses the opportunity to damage the system in the current period but decreases its reputation and weight for causing damage in the future as we explained above for the node with a reputation of 0.01. The two strategies we just described represent the strategy *P* that departs from the strategy *N* in (25). Thus, a malicious node's immediate payoff when playing *N* (25) is higher than when deviating or playing *P*. A malicious node's immediate payoff when playing the strategy *N* is $-D(t)$ while its immediate payoff is $-C(t)$ when playing the strategy *P*. Those payoffs have a negative sign because the game is zero-sum. Any loss to the defender is a win to the malicious node. Recall the defender's payoff is the opposite. We have $-C(t) \leq -D(t)$. The only case in which $-C(t) = -D(t)$ will correspond to a malicious node that has zero weight. That is because it has no influence on the decision process when at least one other node has a weight different from zero. In all other cases, the malicious node has a weight different from zero and we have $-C(t) < -D(t)$.

A malicious node's payoff at any time *t* will depend on the binary strategy adopted (*P* or *N*), the discount factor δ and the smoothing factor γ. Let $V_{\delta\gamma}^{P}(t)$ and $V_{\delta\gamma}^{N}(t)$ denote a malicious node's payoff at time *t* when it chooses to play *P* or *N* respectively. Also, let $V_{\delta\gamma}^{P}(t+1)$ and $V_{\delta\gamma}^{N}(t+1)$ represent a malicious node's future payoffs after playing *P* or *N* respectively.

The Bellman equation [27] allows us to represent a malicious node payoff at any time *t* as a function of its immediate payoff (-*C* or -*D*) and his future payoff. It can be seen that, when there is no noise in recording node's reputation, we must have:

$$\begin{cases} V_{\delta\gamma}^{P}(t) = -(1-\delta)C(t) + \delta V_{\delta\gamma}^{P}(t+1) \\ V_{\delta\gamma}^{N}(t) = -(1-\delta)D(t) + \delta V_{\delta\gamma}^{N}(t+1) \end{cases} \quad (26)$$

At any time a malicious node with a weight different from zero plays *P*, its immediate payoff is lower, but its reputation moves faraway from 0.5 and thus increases its future payoff. On the contrary, at any time a malicious node with a weight different from zero plays *N*, its immediate payoff is higher, but its reputation moves closer to 0.5 and thus decreases its future payoff. Since playing *P* allows a node to accumulate a reputation, (positively or negatively) and increase its potential for future damage, then for a malicious node with a weight different from zero, we must have:

$$V_{\delta\gamma}^{P}(t+1) > V_{\delta\gamma}^{N}(t+1). \quad (27)$$

We have $V_{\delta\gamma}^{P}(t+1) = V_{\delta\gamma}^{N}(t+1)$ if and only if a node has zero weight at time *t*. That is because its future reputation after playing *P* or *N* will be symmetric around 0.5 and thus yields the same effect as explained above in the case 0.99 and 0.01. Note that (27) holds regardless of the continuation strategy. We can see from (26) that a malicious node must perform an inter-temporal optimization that depends on the discount factor δ.

Theorem 3: If there is at least one regular node with a weight that is strictly positive and the discount factor δ is low, then a malicious node must play according to the strategy in (25) which is its dominant strategy.

Proof: We use one-shot deviation principle of dynamic programming [28]. Then, we just need to show that there is no profitable one-shot deviation. A deviation from (25) is profitable to a malicious node if:

$$V_{\delta\gamma}^{P}(t) > V_{\delta\gamma}^{N}(t) \Rightarrow$$

$$-(1-\delta)C(t) + \delta V_{\delta\gamma}^{P}(t+1) > -(1-\delta)D(t) + \delta V_{\delta\gamma}^{N}(t+1)$$

$$\Rightarrow V_{\delta\gamma}^{P}(t+1) > V_{\delta\gamma}^{N}(t+1) + \left(\frac{1-\delta}{\delta}\right)[C(t) - D(t)] \quad (28)$$

We can distinguish two cases. In the first case, the malicious node has a weight different from zero, and then we have $C(t) - D(t) > 0$. Since (27) holds and the right hand side of (28) monotonically grows to infinity as the discount factor is small, there exists $\underline{\delta}$, with $0 < \underline{\delta} < 1$, such that for any $\delta < \underline{\delta}$, (28) does not hold and thus deviation is unprofitable. In the second case, a node with a zero weight does not participate at all in the decision if at least one node has a weight different to zero. This means that $C(t) - D(t) = 0$. Moreover, when a node has zero weight, $V_{\delta\gamma}^{P}(t+1) = V_{\delta\gamma}^{N}(t+1)$. Thus, $V_{\delta\gamma}^{P}(t) = V_{\delta\gamma}^{N}(t)$. Such a node is indifferent from *T* and *F* and thus, is consistent with our strategy.

■

Notice that as opposed to other game theoretic models that require a high discount factor to enforce an equilibrium profile, this work needs a low discount factor. When the discount factor is large $(\underline{\delta} \leq \delta < 1)$, the malicious nodes are tempted to accumulate a reputation for a potential future damage. To prevent this, the defender must divide the game using the framework originally proposed by Ellison [29]. With this framework, the defender divides the game in *M* separate games and record separate reputations for each game. The first game taking place in period *1, M+1, 2M+1,3M+1….* The second in period *2, M+2, 2M+2,…*, and so on. Since the games are separate, the outcome of one game does not influence the outcome of the other game; a malicious node's best response must be independent across the different games.

As a result, the new discount factor in each of the separate game becomes δ^M, which monotonically decreases as *M* increases. Therefore, there exists M_δ such that for all $M > M_\delta, \delta^M < \underline{\delta}$. The defender may choose *M* to be the least greatest integer such that *M* is greater than M_δ up to the time the last regular node is compromised. The more intuitive way to understand the mechanism we just described is that it will take a longer

time for a malicious node to accumulate reputation in each separate game. Say, for instance, that the nodes vote once a day and possibly accumulate a reputation. If the defender divides the game into 365 parts, a malicious node that accumulates any reputation today must wait a year before creating any damage. Therefore, after game separation, it becomes optimal for all compromised nodes to cast bad votes (play *N*) and not accumulate any reputation at all. Also, in our game model, we assume that the defender or system administrator can divide the game as many times as he wants and enforce the strategy in (25) to be the dominant strategy for malicious nodes for any discount factor. Thus, we have the following theorem.

Theorem 4: If there is at least one regular node with a weight that is strictly positive, then a malicious node must play according to the strategy in (25) which is its dominant strategy for any discount factor δ.

Proof: The proof of this theorem derives from the proof of Theorem 3 and Ellison [29 (lemma 2, pg 586)].

■

There are four main advantages for the defender when the malicious nodes are rationally constrained to play the strategy above (25). First, this strategy ensures that the malicious nodes always have a weight close to zero and then never participate in the decision process when there is at least one regular node. Second, the system can survive as long as there is at least one regular node. This is opposed to the single-shot voting game in which system survivability ends with the compromising of a majority of nodes. The extension of the mission survival time under malicious nodes intervention, which constitutes one of the principal goals of this paper, can then be achieved by the defender's rule in Section VI-A. Third, there is a separating equilibrium, so regular nodes vote truthfully while the malicious nodes play the strategy (25). Fourth, malicious nodes eliminate themselves from the decision process.

C. Repeated Voting Game under Imperfect Monitoring

We now consider noise in the defender's detection of the malicious nodes' vote. For instance, a malicious node may play *T* while the defender observes *F*. Therefore, we have an imperfect monitoring game. There is a signal $y \in \{\overline{y}, \underline{y}\}$ indicating the past vote of a node.

Let ε denotes the error in the defender's vote recording. The distribution of the signal *y* is given by:

$$m(\overline{y}|\text{vote}) = \begin{cases} 1-\varepsilon, \text{if a vote is correctly recorded} \\ \varepsilon, \text{if a vote is incorrectly recorded} \end{cases} \quad (29)$$

We assume that $0 < \varepsilon < 1 - \varepsilon < 1$. This ensures that it is more likely that the defender observes the high quality signal $\overline{y}$ when a malicious node vote correctly and also observes the low quality signal $\underline{y}$ otherwise. It should not be confused with noise when the nodes detect the state of nature, nor should it be confused with noise when the defender records a node' vote according to the algorithm in Table II. We are dealing here with the second type of noise.

The strategy we describe in (25) cannot directly be applied as a node may not know its exact reputation. However, a malicious node can update its belief about its reputation and weight over time according to Bayes Rule [28] and play (25) while conditioning its action on its belief about its weight instead of its true weight. In fact, the defender can also constraint the malicious nodes to play such a strategy under noise. This follows from the argument of theorem 3 and 4. To see why, under noise, (26) is transformed as below

$$\begin{cases} V_{\delta\gamma}^{P}(t) = -(1-\delta)C(t) + \delta\left[(1-\varepsilon)V_{\delta\gamma}^{P}(t+1) + \varepsilon V_{\delta\gamma}^{N}(t+1)\right] \\ V_{\delta\gamma}^{N}(t) = -(1-\delta)D(t) + \delta\left[\varepsilon V_{\delta\gamma}^{P}(t+1) + (1-\varepsilon)V_{\delta\gamma}^{N}(t+1)\right] \end{cases}$$

After calculation, a deviation is profitable if:

$$V_{\delta\gamma}^{P}(t+1) > V_{\delta\gamma}^{N}(t+1) + \left(\frac{1-\delta}{\delta}\right)\frac{[C(t)-D(t)]}{(1-2\varepsilon)} \quad (30)$$

Equation (30) is similar to (28) and then all arguments developed Theorem 3 and 4 hold when $\varepsilon < 0.5$.

D. Analysis of Survivability Improvement

The model of Section V holds. The time to compromise a node has an exponential distribution. The node failures are assumed to be statistically independent. The probability distribution of the number of compromised nodes is given by (14). Under repeated interactions as described in this section, the aggregate decision can survive when there is at least one regular node. This is because the regular node will more likely vote correctly, have a positive weight, while all the malicious nodes have a weight close to zero after a short time. From (14), the probability that there is at least one regular node among the *N* node is $1-(1-e^{-rt})^{N}$. The aggregate decision survives the malicious nodes if that probability is greater than 0.5. This means:

$$1-(1-e^{-rt})^{N} > 0.5. \quad (31)$$

After all calculation, (31) gives

$$t < \frac{-\ln\left(1-e^{-\frac{\ln 2}{N}}\right)}{r}. \quad (32)$$

This is a huge improvement in the maximum mission times compared to (19).

$$t < \frac{\ln 2p}{r}. \quad (19)$$

As an additional strategy, the defender can increase the number of nodes to improve its survival time. Equation (32) compares to (19) when only one node is used. We can see that if $N = 1$ in (32) and $p = 1$ in (19), then (19) and (32) become identical. Above one node, (32) and then repeated votes are always superior.

We did not emphasize the optimum decision rule of (21) in Section IV because it is equivalent to a simple majority rule if all the nodes have the same weight. However, repeated interactions allow us to build trust - a metric that enables distinguishing of voter competence.

Finally, since newly-compromised nodes have the same reputation as regular nodes, the defender may periodically request phony votes to accelerate the dissipation of that reputation by (20).

VII. Simmulation Results

This section contains MATLAB simulations to support the different game-theoretic techniques analyzed in this paper. Notice that this work has proposed a high level game-theoretic modeling of decision survivability in cyberspace based on binary voting mechanism. Therefore, our model is more reusable and has a broad scope of application. The specific value we have used in this MATLAB simulation is just to illustrate a few specific scenarios. In general, performance results will depend on the specific implementation that in turn depends on a particular network. We examine the changes in system reliability and equilibrium behavior over time based on our dynamic analysis of Section V. We also compare our one-shot game to our repeated game.

Figure 1 summarizes the change in the attacker's behavior in the one-shot game of Section IV. Three nodes are used for illustration with the node reliability chosen to be $p = 0.8$. Nevertheless, our model is scalable. Thus a larger number of nodes can be used with any reliability p in the range $0.5 < p \leq 1$. Initially, there is no compromised node and the defender chooses the majority opinion. Over time, nodes are compromised according to our dynamic analysis of Section V.

At the beginning, any compromised node plays F, with certainty, up to time 0.47 as indicated by (19). We have chosen $r = 1$ compromising per month. One month is just for illustration. It makes no difference if the time unit is changed to a week or a year.

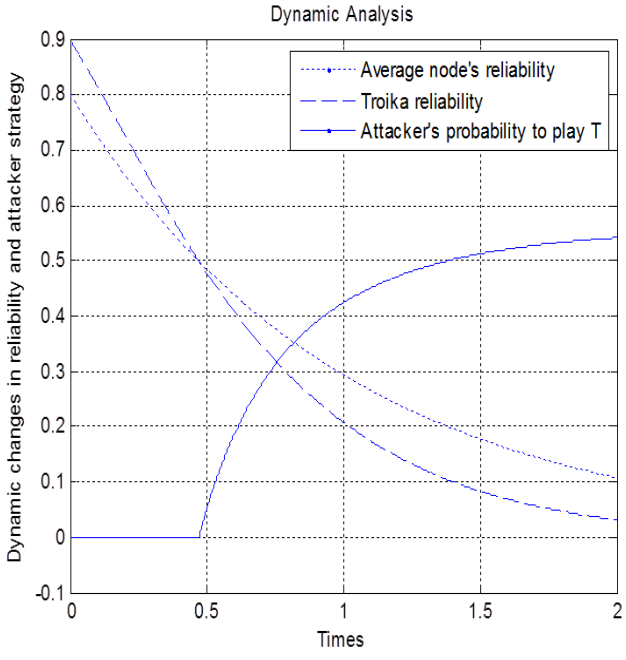

Figure 1: Attacker behavior in the one-shot voting game

After the time 0.47, where the three nodes reliability falls below 0.5, both the attacker and the defender start their mixed strategy. To avoid being exploited by the defender, the probability at which the attacker plays T increases over time. In fact, that probability must even be larger as node reliability decreases.

Figure 2 shows our repeated game model simulated with 5 nodes. Malicious nodes are considered to be very patient with a corresponding discount factor of $\delta = 0.99$. As described in Section VI, such a high discount factor is among the worst case scenarios for the defender. Node reputation is updated using a smoothing factor of $\gamma = 0.1$. As indicated in Section VI, the defender should choose a smoothing factor γ that is neither too large nor too small. A very large smoothing factor disproportionally allocates more weight to the last action whereas a too small smoothing factor will slow the increase of the reputation of regular nodes. A tradeoff value of the smoothing factor may be chosen around 0.1 to balance those opposing factors. Recall that choosing a smoothing factor is totally under the control of the defender. The 5 nodes have the same reliability $p = 0.9$. We consider that every 10 minutes there is an event for which the nodes are requested to vote. We make the conjecture that, with a large discount factor of 0.99, the inequality in (28) holds and thus it is optimum for malicious nodes to accumulate a reputation for future attack. To prevent that, the defender divides the game in 432 parts ($M = 432$) as explained in Subsection VI-B. In each separate game, the nodes vote every three days. The new discount factor, after game separation, becomes $\delta^M = 0.99^{432} = 0.01313$ which is quite low. As a consequence, all malicious nodes are forced to play our strategy represented in (25). The time to compromise a node is exponentially distributed with mean one month or 30 days. In our MATLAB simulation, nodes 1 to 5 are compromised at time (in days) 93.399, 42.468, 24.465, 11.229, and 8.433 respectively. This is only a specific scenario generated according to an exponential distribution with mean 30 days. Different simulations may yield different values but the trend presented here will be preserved. Observe that a node's behavior over time goes through three phases. In the first phase, a node is not yet compromised so its reputation globally increases (with the possibility of occasional decrease because nodes are not perfectly reliable). In the second phase, the node has just become compromised and misbehaves, so its weight decreases to a point slightly below zero. In the third phase, the node's weight alternately goes up and down around zero. Figure 2 illustrates all of those phases for each of these nodes. Figure 2 also shows the aggregate decision. A 1 (one) represents a correct aggregate decision while a 0 (zero) show that the decision is incorrect. We can see in this simulation that, from day 60 to 93, the aggregate decision remains correct although 4 out of 5 nodes are compromised. The aggregate decision actually survives until the last node is compromised.

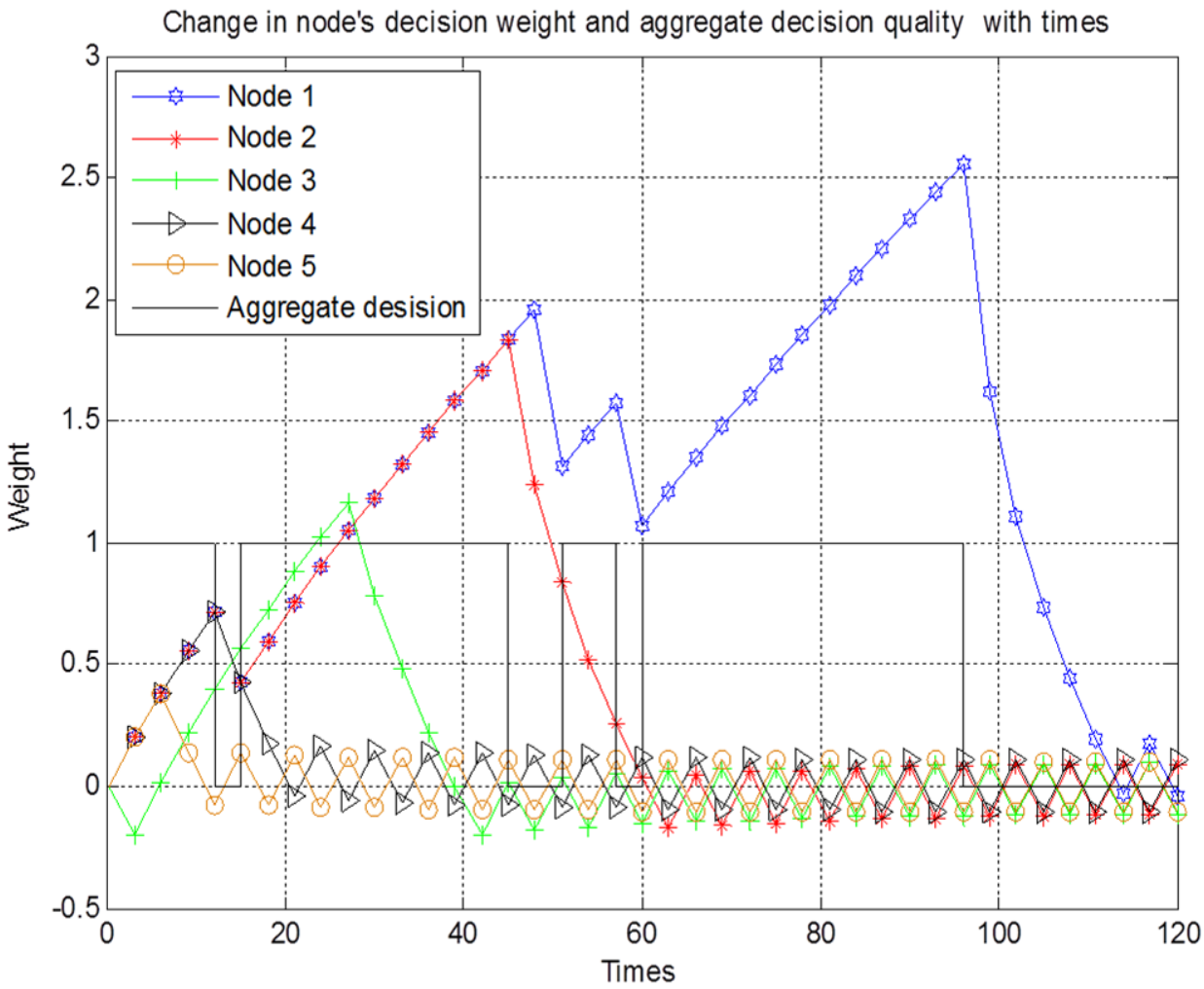

Figure 2: Repeated game simulation with five nodes

Figure 2 also shows that the defender must choose an appropriate smoothing factor γ (20). A very low smoothing factor will slow the increase in reputation (and weight) of regular nodes in the first phase. Moreover, the reputation of compromised nodes in the second phase will decrease slowly. However, with a very high smoothing factor, the weight of compromised nodes will be relatively high in the third phase. Figure 2 also shows that in addition to intrusion resilience, our model can achieve intrusion detection by looking into the behavior of the nodes that have a weight up and down around zero.

One limitation of our model is the case in which the nodes have a very low mean time to compromise. When the mean time to compromise is very low, the entire node can be compromised before any regular node has time to build any reputation.

The maximum mission times (or the time before which the votes can give any information to the defender) are compared in the two scenarios we have evaluated: the one-shot voting game and repeated voting game. Figure 3 shows that the mission survival time is always superior in the repeated game model. This comparison was done with only three nodes. The difference dramatically increases as the number of node increases. This is because, in the one-shot game, the mission survival time does not change with the number of nodes while it does increase in repeated game. As you can see, the simulation results are encouraging for implementation in a mission essential function in cyberspace.

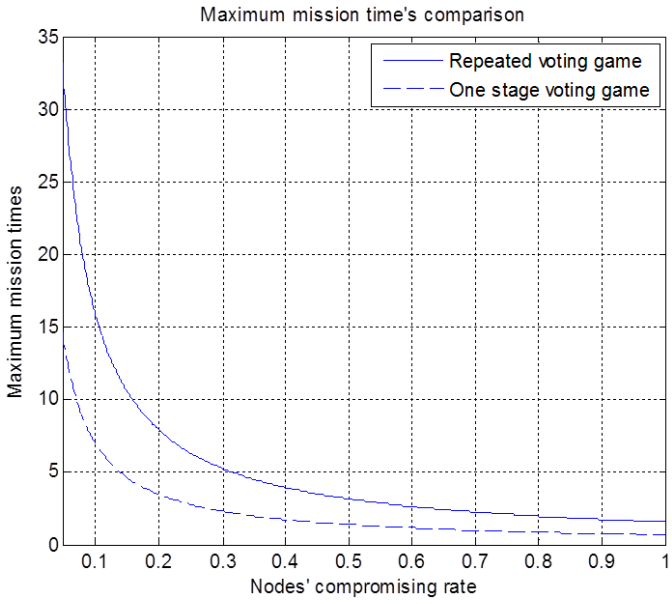

Figure 3: Comparison of the maximum mission times

VIII. Conclusion and Future Works

This research has used several approaches from game theory and mechanism design to support binary decision survival subject to influence from malicious nodes. The prospect of the paper was oriented toward a scenario pitting a network attacker against a network defender. In most cases, a failure rate is completely ignored. Our models capture the failure dynamic of attacker-induced faults in real time. Recognizing that any attempt to consider only benign node failures will be defeated when subjected to the maliciousness of intelligent attacks, we have restricted the scope of this paper to such malicious attacks. We have proposed two models, a one-shot game model and a repeated game model. Our one-shot game demonstrates a fundamental property of binary game. If the number of compromised nodes is above some quota, (generally less than 50% of the nodes) the maximum decision reliability will be 50%. This means in the context of binary vote that no useful information can be derived from the votes. Our repeated game model overcomes this intrinsic deficiency, by proposing a model in which *the aggregate vote reliability stays above 50% even though nearly all the nodes are compromised*. That is because regular nodes have a higher weight in the aggregate decision. The defender designs, implements, and announces the rules of the repeated game. Those rules are such that all the malicious nodes are squeezed toward behaviors that are fault tolerable. In the extreme case, a single regular node may surpass the vote of hundreds of malicious nodes. For example, in a set of 115 nodes, a single node with a reputation of 0.99 cannot have its vote overturn by 114 compromised nodes having a reputation of 0.51. Our repeated game model considers two elements: a node's reliability and a node's rate of being compromised. From these elements, a three-part mechanism - totally controlled by the defender - ensures a high decision reliability. The first part of our mechanism is an exponentially weighted moving average to accurately update the node reputation according to the most recent behavior. The second part is a mathematically proven optimum weight derived from the node's reputation. The third part is a game separation method that discourages malicious nodes to accumulate any reputation or have any weight in the decision process. Our model is supported by mathematical proofs and extensive simulation results. Using the framework described in this paper, a mission can have its failures masked, prolong its survival in cyberspace, and fight through attacks by assuring accuracy of critical decisions in highly-contested environments. The case of correlated nodes' failure will be the subject of our future investigation. Also, the case of possible collusions among the malicious nodes in our repeated game model needs further analysis.

Acknowledgment

This research was performed while Charles Kamhoua and Joon Park held a National Research Council (NRC) Research Associateship Award at the Air Force Research Laboratory (AFRL). This research was supported by the Air Force Office of Scientific Research (AFOSR).

References

[1] K. Kwiat, A. Taylor, W. Zwicker, D. Hill, S. Wetzonis, S. Ren "Analysis of binary voting algorithms for use in fault-tolerant and secure computing" International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt. December 2010.

[2] W. Du, J. Deng, Y. Han, P. Varshney "A withness-based approach for data fusion assurance in wireless sensor networks" IEEE GLOBECOM 2003.

[3] L. Wang, Z. Li, S. Ren, K. Kwiat "Optimal Voting Strategy Against Rational Attackers" The Sixth International Conference on Risks and Security of Internet and Systems CRiSIS 2011, Timisoara, Romania, September 2011.

[4] G. An and J. Park. "Cooperative component testing architecture in collaborating network environment" In Proceedings of the 4th International Conference on Autonomic and Trusted Computing (ATC), Lecture Notes in Computer Science (LNCS), pages 179-190,Hong Kong, China, July 11-13, 2007. Springer.

[5] J. Park, P. Chandramohan, G. Devarajan, J. Giordano. "Trusted component sharing by runtime test and immunization for survivable distributed systems". In Ryoichi Sasaki, Sihan Qing, Eiji Okamoto, and Hiroshi Yoshiura, editors, Security and Privacy in the Age of Ubiquitous Computing, pages127–142. Springer, 2005. Proceedings of the 20th IFIP TC11 International Conference on Information Security (IFIP/SEC), Chiba, Japan, May 30-June 1, 2005.

[6] S. Bhattacharjee, S. Debroy, M. Chatterjee, K. Kwiat "Trust based Fusion over Noisy Channels throught Anomaly Detection in Cognitive Radio Networks" 4th International Conference on Security of Information and Networks (ACM SIN 2011), Sydney, Australia, November 2011.

[7] Z. S. Ma; A.W. Krings, "Dynamic Hybrid Fault Modeling and Extended Evolutionary Game Theory for Reliability, Survivability and Fault Tolerance Analyses" IEEE Transactions on Reliability, vol.60, no.1, pp.180-196, March 2011.

[8] D. Malki, M. Reiter "Byzantine quorum systems" Distributed computer system, pp203-213 1998.

[9] D. Gao, M.K. Reiter, D. Song "Beyond Output Voting: Detecting Compromised Replicas Using HMM-Based Behavioral Distance", IEEE Transactions on Dependable and Secure Computing, vol.6, no.2, pp.96-110, April-June 2009.

[10] C. Kamhoua, N. Pissinou, K. Makki " Game Theoretic Modeling and Evolution of Trust in Autonomous Multi-hop Networks: Application to Network Security and Privacy" in proceedings of the IEEE international conference on communications (IEEE ICC 2011). Kyoto, Japan. June 2011.

[11] S. Becker, J. Seibert, D. Zage, C. Nita-Rotaru, R. State"Applying Game Theory to Analyze Attacks and Defenses in Virtual Coordinate Systems" IEEE DSN 2011.

[12] F. Li; J. Wu; "Hit and Run: A Bayesian Game Between Malicious and Regular Nodes in MANETs" SECON '08. 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,, vol., no., pp.432-440, 16-20 June 2008.

[13] Y. Liu, C. Comaniciu, H. Man "Modeling Misbehavior in Ad Hoc Networks: A Game Theoretic Approach for

Intrusion Detection", International Journal of Security and Networks (IJSN) 2006.
[14] M. Manshaei , Q. Zhu , T. Alpcan, T. Basar, J-P Hubaux "Game Theory Meets Network Security and Privacy" ACM transaction on Computational Logic, Vol. V, No. N, September 2010.
[15] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, Q. Wu; , "A Survey of Game Theory as Applied to Network Security," 43rd Hawaii International Conference on System Sciences (HICSS),, vol., no., pp.1-10, 5-8 Jan. 2010.
[16] Condorcet: "*Essai sur l'application de l'analyse a la probabilite des decisions rendues a la pluralite des voix*". Paris: Imprimerie Royale. 1785.
[17] G. Owen, B. Grofman, S. Feld "Proving a Distribution-Free Generalization of the Condorcet Jury Theorem" Mathematical Social Sciences 17, 1-16, 1989.
[18] R. Myerson "Extended Poisson Games and the Condorcet Jury Theorem" Games and Economic Behavior, Elsevier, 1998.
[19] J-F Laslier, J. W. Weibull "The Condorcet Jury Theorem and Heterogeneity" 2010.
[20] L. Shapley, B. Grofman "Optimizing group Judgemental Accuracy in Presence of Interdependencies" Public Choice 43: 329-343, 1984.
[21] R. Goodin, D. Estlund "The Persuasiveness of Democratic Majorities" Politics, Philosophy & Economics 131-142, 2004.
[22] B. Grofman, G. Owen, S. Feld "Thirteen Theorems in Search of the Truth" Theory and Decision 15, 261-278, 1983.
[23] R. Myerson "Game theory: analysis of conflict" Harvard University Press, 1997.
[24] D. J. Leversage, E. J. Byres "Estimating a System's Mean Time-to-Compromise" IEEE Security & Privacy, 2008
[25] M. A. McQueen, W. F. Boyer, M. A. Flynn, G. A. Beitel "Time-to-Compromise Model for Cyber Risk Reduction Estimation" Quality of Protection, Springer, 2006.
[26] Rescorla, E., "Is Finding Security Holes a Good Idea," IEEE Security & Privacy,January-2005.
[27] D. Bertsekas, "Dynamic Programming and Optimal Control", vol. 1,2, Athena Scientific, Belmont, MA, Second edition, 2001.
[28] G. Mailath, L.Samuelson "Repeated Games and Reputations, Long-run relationships" Oxford university press, 2006.
[29] G. Ellison "Cooperation in the Prisoner's Dilemma with Anonymous Random Matching" The Review of Economic Studies, Vol. 61. No. 3 pp. 567-588, Jul., 1994.

Charles A. Kamhoua received his B.S. in Electronic from the University of Douala/ENSET, Cameroon in 1999. He received his M.S. in Telecommunication and Networking and his PhD in Electrical Engineering from Florida International University in 2008 and 2011 respectively. He is currently a postdoctoral fellow at the Air Force Research Laboratory. His interdisciplinary research area includes game theory, cybersecurity, survivability, fault tolerant networks, and ad hoc networks.

Dr. Kamhoua is a member of IEEE and the National Society of Black Engineer (NSBE). He is the recipient of the National Academies Postdoctoral Fellowship award at the Air Force Research Laboratory, Rome, New York in March 2011, extended in February 2012.

Kevin A. Kwiat is a Principal Computer Engineer in the Cyber Science Branch of the U.S. Air Force Research Laboratory (AFRL) in Rome, New York where he has worked for over 28 years. He received a Ph.D. in Computer Engineering from Syracuse University.

Dr. Kwiat holds 4 patents and has published more than hundred journal and conference papers. In addition to his duties with the Air Force, he is an adjunct professor of Computer Science at the State University of New York at Utica/Rome, an adjunct instructor of Computer Engineering at Syracuse University, and a Research Associate Professor with the University at Buffalo.

Joon S. Park is an associate professor at the School of Information Studies (iSchool), Syracuse University, Syracuse, New York, USA. Currently, he is the director of the Certificate of Advanced Study (CAS) in Information Security Management (ISM) at the iSchool.

Dr. Park has been involved with research/education in information and systems security over the past decades. Before he joined the iSchool, he worked for the Center for High Assurance Computer Systems (CHACS) at the U.S. Naval Research Laboratory (NRL), Washington, D.C. He received a PhD in Information Technology, specialized in Information Security, from George Mason University, Fairfax, Virginia, in 1999.