An Adaptive Markov Game Model for Cyber Threat Intent Inference

Dan Shen[1], Genshe Chen[2], Jose B. Cruz, Jr.[3], Erik Blasch[4], and Khanh Pham[4]
[1]Intelligent Automation, Inc.,
[2]DCM Research Resources, LLC.
[3]The Ohio State University,
[4]The Air Force Research Laboratory,
USA

1. Introduction

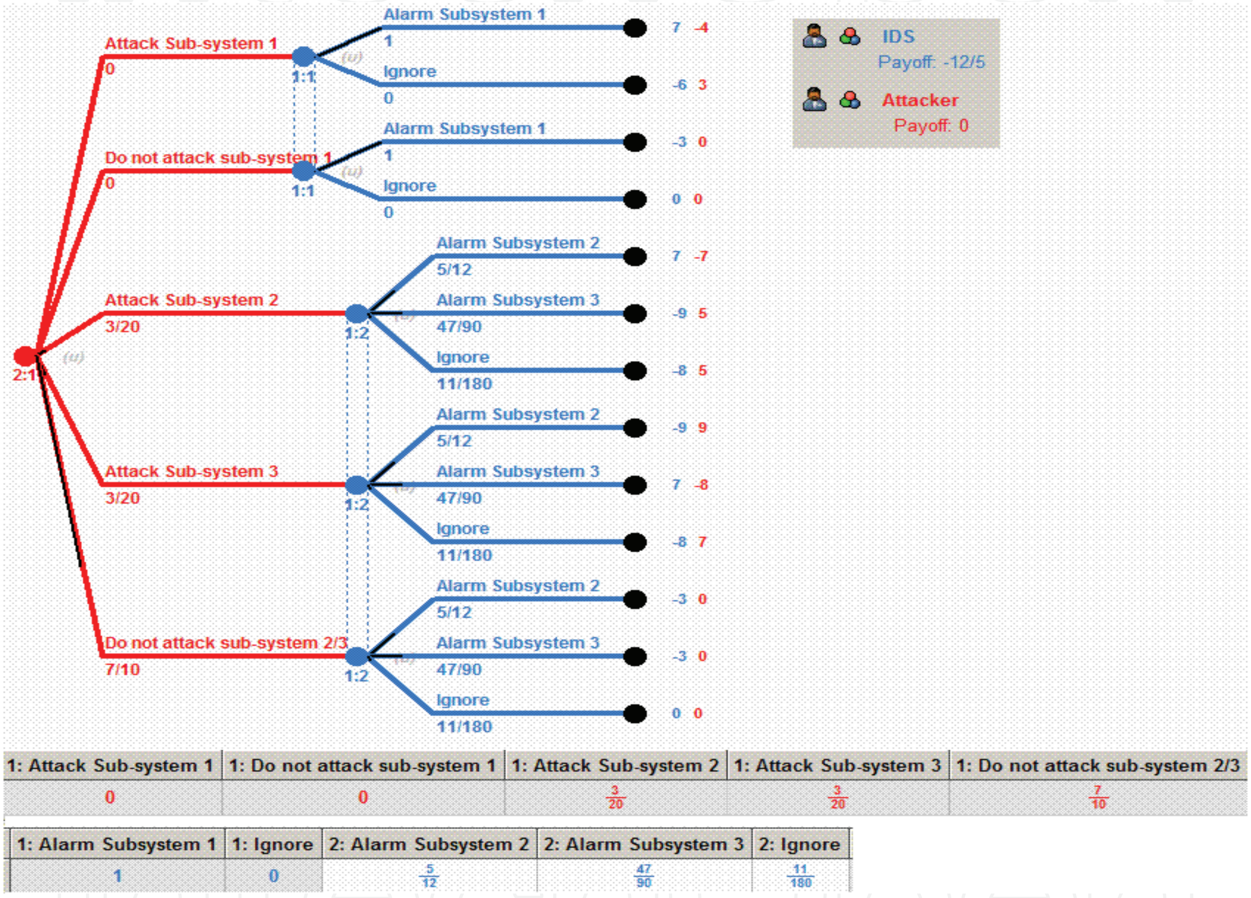
Cyber attacks (CAs) have generally been one-dimensional, involving denial of service (DoS), computer viruses or worms, and unauthorized intrusion (hacking). Websites, mail servers, and client machines are the major targets. However, recent CAs have diversified to include multi-stage and multi-dimensional attacks with a variety of tools and technologies. Next-generation security will require network management and intrusion detection systems that combine short-term sensor information with long-term knowledge databases to provide decision support and cyberspace command and control. One of the important capabilities is to efficiently and promptly predict the threat's tactical intent from various network alerts generated by Intrusion Detection Sensors (IDSs) or Intrusion Prevention Sensors (IPSs).

Recent efforts to apply data fusion techniques to cyber situational awareness are promising (Salerno et al., 2005; Tadda et al., 2006), but assessing the potential impact of an attack and predicting intent, or high-level data fusion, continue to present substantive challenges. In this chapter, an adaptive Markov game approach is introduced to meet the challenge.

Game theory is not a new concept in the cyber defense domain. Current game theoretic approaches (Alpcan & Basar, 2003; Agah et al., 2004; Sallhammar et al., 2005) for cyber network intrusion detection and decision support are based on static matrix games and simple extensive games, which are usually solved by game trees. For example (see Fig. 1), red side (attacker) has five options while blue side (IDS) has two re-actions for information-set 1 (two blue bullets labeled as 1:1) about sub-system 1 and three possible actions for information-set 2 (three blue bullets labeled as 1:2) about sub-system 2 and 3. The payoffs of both sides are shown on the right side of each possible outcome (black bullets). A mixed Nash Strategy pair is shown with black lines. Attacker will choose action "Attack Sub-system 1" with probability 3/20, "Attack Sub-system 2" with probability 3/20, and "Do not attack sub-system 2/3" with probability 7/10. Blue side will set an alarm for sub-system 1 with probability 1, sub-system 2 with probability 5/12, sub-system 3 with probability 47/90, and ignore with probability 11/180. It is not difficult to see that these matrix game models lack the sophistication to study multi-players with relatively large actions spaces, and large planning horizons.

Source: Theory and Novel Applications of Machine Learning, Book edited by: Meng Joo Er and Yi Zhou,
ISBN 978-3-902613-55-4, pp. 376, February 2009, I-Tech, Vienna, Austria

In this chapter, we propose a game theoretic situation awareness and impact assessment approach for cyber network defense system to consider the changes of threat intents during cyber conflict. From a perspective of data fusion and adaptive control, we use a Markov (stochastic) game method to estimate the belief of each possible cyber attack pattern. With the consideration that the parameters in each game player's cost function is not accessible to other players, we design an adaptation scheme, based on the concept of Fictitious Play (FP), for the piecewise linearized Markov game model. A software tool is developed to demonstrate the performance of the adaptive game theoretic high level information fusion approach for cyber network defense and a simulation example shows the enhanced understating of cyber-network defense.



1: Attack Sub-system 1	1: Do not attack sub-system 1	1: Attack Sub-system 2	1: Attack Sub-system 3	1: Do not attack sub-system 2/3
0	0	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{7}{10}$

1: Alarm Subsystem 1	1: Ignore	2: Alarm Subsystem 2	2: Alarm Subsystem 3	2: Ignore
1	0	$\frac{5}{12}$	$\frac{47}{90}$	$\frac{11}{180}$

Fig. 1. An example of decision support based on static matrix game model

Our approach has the following advantages: (1) it is decentralized. Each network defense element or team makes decisions mostly based on local information. We put more autonomies in each group allowing for more flexibilities; (2) a Markov decision process (MDP) can effectively model the uncertainties in the noisy cyber environment; (3) it is a game model with two players: red force (network attackers) and blue force (cyber defense resources); and (4) FP learning concept is integrated. Each player presumes that his opponents are playing stable strategies (Nash equilibria). Each player starts with some initial beliefs and chooses a best response to those beliefs as a strategy in this round. Then, after observing their opponents' actions, the players update their beliefs. The process is then repeated. It is known that if it converges, then the point of convergence is a Nash equilibrium of the game.

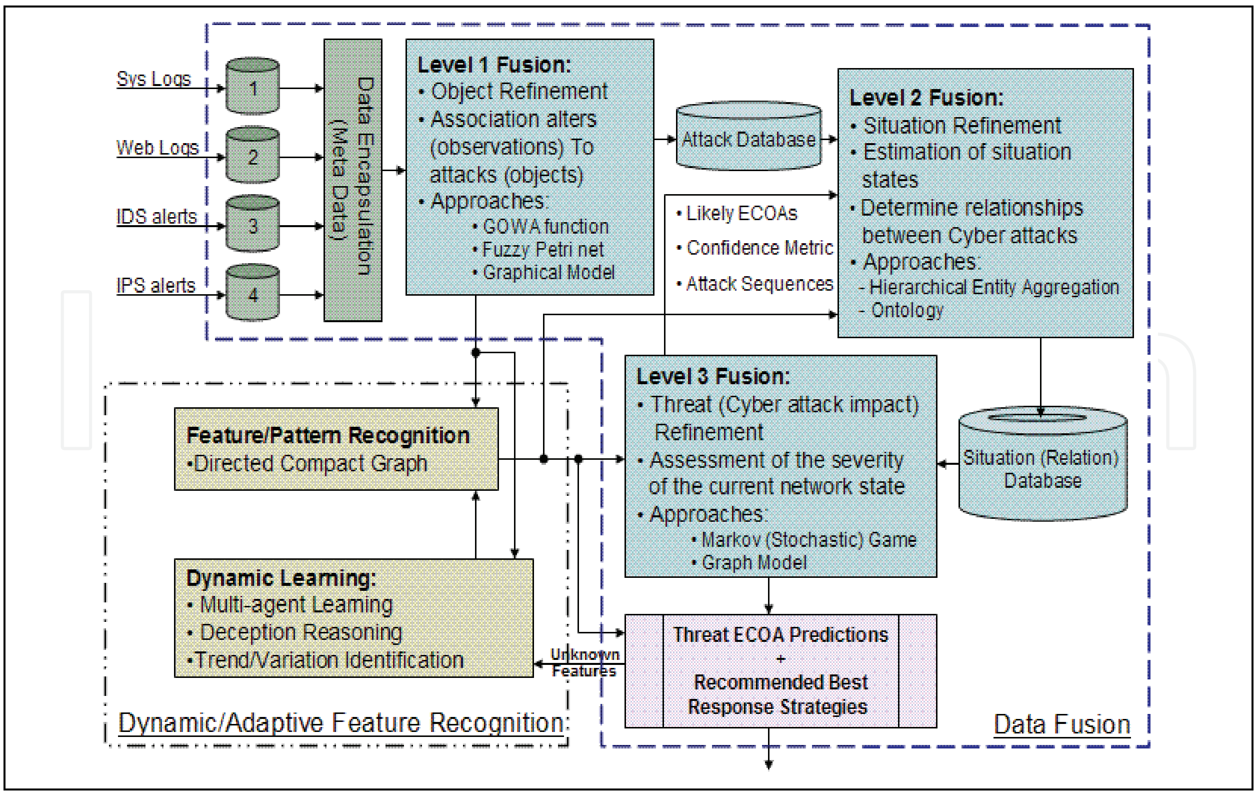
The rest of the chapter is organized as follows. Section 2 describes our proposed framework. Section 3 presents a Markov model for cyber network defense. Then an adaptive design is described in Section 4. Section 5 discusses the simulation tool and simulation result, and Section 6 gives conclusions.

2. Framework for Cyber Threat Intent Inference

As indicated in Fig. 2, our cyberspace security framework has two fully coupled major parts: 1) *Data fusion module* (to refine primitive awareness and assessment, and to identify new cyber attacks); and 2) *Dynamic/adaptive feature recognition module* (to generate primitive estimations, and to learn new identified new or unknown cyber attacks).

The data fusion module permits refinement of primitive awareness and assessment to identification of new attacks while the dynamic/adaptive feature recognition module generates estimates and learns about them. The adaptive Markov game method, a stochastic approach, is used to evaluate the prospects of each potential attack. Game theory captures the nature of cyber conflict: determining the attacker's strategies is closely allied to decisions on defense and vice versa.

Fig. 2 also charts the data mining and fusion structure. For instance, detection of new attack patterns is linked to Level-1 (L1) data fusion results in dynamic learning, including deception reasoning, trend/variation identification, and multi-agent learning. Our approach to deception detection is heavily rooted in the application of pattern-recognition techniques to locate and diagnose anomalous conditions in the cyber environment. Dynamic learning and refinement can also enhance Level-2 (L2) and Level-3 (L3) data fusion.



Sys Logs
Web Logs
IDS alerts
IPS alerts
1
2
3
4
Data Encapsulation (Meta Data)
Level 1 Fusion:
• Object Refinement
• Association alters (observations) To attacks (objects)
• Approaches:
• GOWA function
• Fuzzy Petri net
• Graphical Model
Attack Database
Level 2 Fusion:
• Situation Refinement
• Estimation of situation states
• Determine relationships between Cyber attacks
• Approaches:
- Hierarchical Entity Aggregation
- Ontology
• Likely ECOAs
• Confidence Metric
• Attack Sequences
Level 3 Fusion:
• Threat (Cyber attack impact) Refinement
• Assessment of the severity of the current network state
• Approaches:
• Markov (Stochastic) Game
• Graph Model
Situation (Relation) Database
Feature/Pattern Recognition
•Directed Compact Graph
Dynamic Learning:
• Multi-agent Learning
• Deception Reasoning
•Trend/Variation Identification
Unknown Features
Threat ECOA Predictions
+
Recommended Best Response Strategies
Dynamic/Adaptive Feature Recognition
Data Fusion

Fig. 2. Data Fusion Approach for Cyber Situation Awareness and Impact Assessment

Various logs and alerts generated by Intrusion Detection Sensors (IDSs) or Intrusion Prevention Sensors (IPSs) are fed into the L1 data fusion components. The fused objects and related pedigree information are used by a feature/pattern recognition module to generate primitive prediction of intents of cyber attackers. If the observed features are already associated with adversary intents, we can easily obtain them by pattern recognition. In some time-critical applications, the primitive prediction can be used before it is refined; because the high-level data fusion refinement operation is relatively time-consuming in the multiplicative of probability calculations.

High-level (L2 and L3) data fusion based on Markov game model is proposed to refine the primitive prediction generated in stage 1 and capture new or unknown cyber attacks. The adaptive Markov (Stochastic) game method (AMGM) is used to estimate the belief of each possible cyber attack graph. Game theory can capture the nature of cyber conflicts: the determination of the attacking-force strategies is tightly coupled to the determination of the defense-force strategies and vice versa. Also AMGM can deal with the uncertainty and incompleteness of the available information. We propose a graphical model to represent the structure and evolution of the above-mentioned Markov game model so that we can efficiently solve the graphical game problem.

The captured unknown or new cyber attack patterns will be associated to related L1 results in the dynamic learning block, which takes deception reasoning, trend/variation identification, and distribution models and calculations into account. Our approach to deception detection is heavily based on the application of pattern recognition techniques to detect and diagnose what we call out-of-normal (anomaly) conditions in the cyber environment. The results of dynamic learning or refinement shall also be used to enhance L2 and L3 data fusion. This adaptive process may be considered as level 4 data fusion (process refinement; see the 2004 DFIG model (Blasch & Plano, 2005)).

In this chapter, we will focus on the L3 data fusion (adaptive Markov game approach) part in the overall framework shown in Fig. 2.

3. Markov game model for cyber network defense

In general, a Markov (stochastic) game (Shapley, 1953) is specified by (i) a finite set of players *N*, (ii) a set of states *S*, (iii) for every player $i \in N$, a finite set of available actions D^i (we denote the overall action space $D = \times_{i \in N} D^i$), (iv) a transition rule $q : S \times D \to \Delta(S)$, (where $\Delta(S)$ is the space of all probability distributions over *S*), and (v) a payoff function $r : S \times D \to R^N$. For the cyber decision support and attacker intent inference problem, we obtain the following distributed discrete time Markov game (we revise the Markov game model (Chen et al., 2007) used for battle-space and focus on the cyber attack domain properties):

Players (Decision Makers) --- Cyber attackers and network defense system are two players of this Markov game model. We denote cyber attackers as "red team", and the network defense system (IDSs, Firewalls, Email-Filters, Encryption) as "blue team". The cooperation within the same team is also modeled so that the coordinated cyber network attacks can be captured and predicted.

State Space --- All the possible states of involved network nodes consist of the state space. For example, the web-server (IP = 26.134.3.125) is controlled by attackers. To determine the optimal IDS deployment, we include the defense status for each network nodes in the state space. So, for the i^{th} network node, there is a state vector $s^i(k)$ at time *k*.

$$s^i(k) = (f, p, a)^T \tag{1}$$

where *f* is the working status of the *i*th network node, *p* is the protection status, *a* is the status of being attacked, and *T* is the transpose operator. "Normal" and "malfunction" are typical values of *f* with the meaning that the node is in the normal working status or malfunction (Recall that in battle space cases, the function status of any unit values can be "undestroyed", "damaged", or "destroyed"). *p* can be the defense unit/service (such as firewall, IDS and filter, with probability) assigned to the node and *p* = NULL means that the *i*th node is unprotected. The type of attacks will be specified in Action Space.

Remark 1: *It is not difficult to understand that the system states are determined by two factors: 1) previous states and 2) the current actions. So the whole system can be modelled by a first-order Markov decision process.*

The overall system state at time k is

$$s_k = [s^1(k), s^2(k), \cdots, s^M(k)] \tag{2}$$

where *M* is the number of nodes in the involved cyber network.

Remark 2: *The system states are updated based on the IDS and the control inputs from the decision makers. Our model does not require 100% accurate measurements from the IDS. The IDS can report several possible situations with the associated confidence vector. For example, one IDS can input "node 1 is damaged with the probability 0.85" and "node 1 is normal with probability 0.15".*

Action Space --- At every time step, each player chooses targets with associated actions based on its observed network information. For the red team (cyber network attackers), we consider the following types of network-based attacks:

Buffer overflow (web attack): it occurs when a program does not check to make sure the data it is putting into a space will actually fit into that space.

Semantic URL attack (web attack): In semantic URL attack, a client manually adjusts the parameters of its request by maintaining the URL's syntax but altering its semantic meaning. This attack is primarily used against Common Gateway Interface (CGI) driven websites.

E-mail Bombing (email attack): In Internet usage, an e-mail bomb is a form of net abuse consisting of sending huge e-mail volumes to an address in an attempt to overflow the mailbox or overwhelm the server.

E-mail spam (email attack): Spamming is the abuse of electronic messaging systems to send unsolicited, and/or undesired bulk messages.

MALware attachment (email attack): Malware is software designed to infiltrate or damage a computer system without the owner's informed consent. Some common MALware attacks are worms, viruses, Trojan horses, etc.

Denial-of-service (network attack): Denial-of-service (DoS) attack is an attempt to make a computer resource unavailable to its intended users. Typically the targets are high-profile web servers where the attack is aiming to cause the hosted web pages to be unavailable on the Internet. A distributed denial of service attack (DDoS) occurs when multiple compromised systems flood the bandwidth or resources of a targeted system, usually a web server(s). These systems are compromised by attackers using a variety of methods.

Remark 3: *Some attacks may be multi-stage. For example, e-mail spam and MALware are used first to gain control of several temporal network nodes, which are usually not well protected servers. Then a DoS attack will be triggered to a specified and ultimate target. Our dynamic Markov game model* *can handle these attacks from a planning perspective. Our mixed Nash strategy pair is based on a fixed finite planning horizon. See Strategies for details.*

For the blue team (network defense system), we consider the following defense actions:

IDS deployment: we assume that there are limited IDSs. IDS deployment is similar to resource allocation (target selection) problems in traditional battle-space situations. We try to find an optimal deployment strategy to maximize the chance of detecting all possible cyber network intrusions.

Firewall configuration: A firewall is an information technology (IT) security device which is configured to permit, deny or proxy data connections set and configured by the organization's security policy.

Email-filter configuration: Email filtering is the processing of organizing emails according to a specified criterion. Email filtering software inputs email and for its output, it might (a) pass the message through unchanged for delivery to the user's mailbox, (b) redirect the message for delivery elsewhere, or (c) throw the message away. Some e-mail filters are able to edit messages during processing.

Shut down or reset servers.

Transition Rule --- The objective of the transition rule is to calculate the probability distribution over the state space $q(s_{k+1}|s_k, u_k^B, u_k^R)$, where s_k, s_{k+1} are system states at time *k* and *k*+1 respectively, u_k^B, u_k^R are the overall decisions of the blue team (network defense system) and the red team (cyber attackers), respectively, at time step *k*. How to decide the overall actions for each team are specified in Strategies.

For each network node (server or workstation), the state of time *k*+1 is determined by three things: 1) state at time *k*; 2) control strategies of the two players; and 3) the attack/defense efficiency. If we compare part 3) to battle-space domain, the efficiency is the analogue of kill probability of weapons.

For example, if the state of node 1 at time k is [“normal”, “NULL”, “NULL”], one component of the red action is “email-bombing node 1”, and one component of blue action is “email-filter–configuration-no-block for node 1”, then the probability distribution of all possible next states of node 1 is: [“normal”, “email-filter-configuration”, “email-bombing”] with probability 0.4; [“slow response”, “email-filter-configuration”, “email-bombing”] with probability 0.3; and [“crashed”, “email-filter-configuration”, “email-bombing”] with probability 0.3. The actual probabilities depend on the efficiency of attacking and defending actions.

Payoff Functions --- In this Markov game model, there are two levels of payoff functions for each player (red or blue): lower-level (cooperative within each team) and higher-level (non-cooperative between teams) payoff functions. This hierarchical structure is important to model the coordinated cyber network attacks and specify optimal coordinated network defense strategies and IDS deployment.

The lower level payoff functions are used by each team (blue or red) to determine the cooperative team actions for each team member based on the available local information. For the *j*th unit of blue force, the payoff function at time *k* is defined as $\phi_j^B\left(\tilde{s}_j^B(k), u_j^B(k), W^B(k); k\right)$, where $\tilde{s}_j^B(k) \subseteq s_k$ is the local information obtained by the *j*th blue member, $u_j^B(k)$ is the action taken by the blue team member at time k, and $W^B(k)$, the weights for all possible action-target couples of blue force, is announced to all blue team members and determined according to the top level payoff functions from a team-optimal perspective.

$$\phi_j^B\left(\tilde{s}_j^B(k),u_j^B(k),W^B(k);k\right) = U\left(\tilde{s}_j^B(k)\right)-w\left(W^B(k),u_j^B(k)\right)C\left(u_j^B(k)\right) \tag{3}$$

where, $U\left(\tilde{s}_j^B(k)\right)$ is the utility or payoff of the current local network state. $C\left(u_j^B(k)\right)$ is the cost of action to be taken by the blue team member. Usually, $U(\cdot)$ is a negative value if a network node is in malfunction status due to a cyber attack. The specific value depends on the value of the network node. The counterpart in the battle-space domain is the target value of each platform. Function $w\left(W^B(k),u_j^B(k)\right)$ will calculate the weight for any specified action decision for the jth member of the blue team based on the received $W^B(k)$, which is determined on a team level and indicates the preference and trend of team defense strategies.

Similarly, we obtain the lower level payoff functions for the jth member of red player,

$$\phi_j^R\left(\tilde{s}_j^R(k),u_j^R(k),W^R(k);k\right) = U\left(\tilde{s}_j^R(k)\right)-w\left(W^R(k),u_j^R(k)\right)C\left(u_j^R(k)\right) \tag{4}$$

The top level payoff functions at time k are used to evaluate the overall performance, $V(\cdot)$, of each team.

$$V^B(\tilde{s}^B(k),u_k^B;k)=\sum_{j=1}^{M^B}\phi_j^B\left(\tilde{s}_j^B(k),u_j^B(k),W^B(k);k\right) \tag{5}$$

$$V^R(\tilde{s}^R(k),u_k^R;k)=\sum_{j=1}^{M^R}\phi_j^R\left(\tilde{s}_j^R(k),u_j^R(k),W^R(k);k\right) \tag{6}$$

In our approach, the lower level payoffs are calculated distributedly by each team member and sent back to network administrator via communication networks.

Strategies --- In game theory, the Nash equilibrium (Nash, 1951) is a kind of optimal collective strategy in a game involving two or more players, where no player has anything to gain by changing only his or her own strategy. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium.

In our proposed approach, the sub-optimal Nash solution to the Markov game is obtained via a K time-step look-ahead approach, in which we only optimize the solution in the K time-step horizon. K usually takes 2, 3, 4, or 5. The suboptimal technique is used successfully for reasoning in games such as chess, backgammon, and monopoly. For our case, the objective of each team at time k is to find a serial of actions or course of action (COA) to maximize the following team level payoffs, $J(\cdot)$, respectively,

$$J_k^B(u_k^B,u_k^R,u_{k+1}^B,u_{k+1}^R,\ldots,u_{k+K}^B,u_{k+K}^R)=\sum_{i=k}^{k+K}V^B(\tilde{s}^B(i),u_i^B;i) \tag{7}$$

$$J_k^R(u_k^B, u_k^R, u_{k+1}^B, u_{k+1}^R, \dots, u_{k+K}^B, u_{k+K}^R) = \sum_{i=k}^{k+K} V^R(\tilde{s}^R(i), u_i^R; i) \tag{8}$$

Remark 4: *The K-step look-ahead (or moving window) approach well fits the situations in which multi-step cyber network attacks occurs since we evaluate the performance of each team based on the sum of payoffs during a period of K-time steps. With bigger K, the game model can predicate or detect more complicated network attackers with more stages. The cost is the increased computation complexity.*

In general, the system model is nonlinear. By the linearization transformation method (Sastry, 1999), the system dynamics can be approximately modeled by a linear system.

$$X_{k+1} = A_k X_k + D_k^B C_k^B + D_k^R C_k^R \tag{9}$$

where X_k is the state of network at time *k*. C_k^B and C_k^R are the action control of blue and red force, respectively, at time k. D_k^B and D_k^R are the related control gains.

The objective of each side is to minimize its cost function

$$J^B = \underbrace{(X_N)^T Q_N^{BR} X_N}_{\text{Gain of Red side}} - \underbrace{(X_N)^T Q_N^{BB} X_N}_{\text{Gain of Blue side}} - \underbrace{\sum_{k=1}^{N-1}\left(\left(C_k^B\right)^T P^B C_k^B\right)}_{\text{Cost of Blue Actions}} \tag{10}$$

$$J^R = \underbrace{(X_N)^T Q_N^{RB} X_N}_{\text{Gain of Blue side}} - \underbrace{(X_N)^T Q_N^{RR} X_N}_{\text{Gain of Red side}} - \underbrace{\sum_{k=1}^{N-1}\left(\left(C_k^R\right)^T P^R C_k^R\right)}_{\text{Cost of Red Actions}} \tag{11}$$

where Q_k^{BR}, Q_k^{BB}, Q_k^{RB}, and Q_k^{RR} are the gain matrices. Each player determines its gain matrix based on the network topology and its goals. P^B and P^R are the cost matrix of blue player and red player, respectively.

4. An adaptation design for linear quadratic games

With the consideration that the parameters in each side's cost function is not accessible to the other side, we propose to use an adaptation design based on the concept of Fictitious Play (FP) (Brown, 1951; Fudenberg & Levine, 1998) to learn these unknown properties. As a learning concept, FP was first introduced by G. W. Brown in 1951. Within the learning scheme, each player presumes that her opponents are playing stable (possibly mixed) strategies. Each player starts with some initial beliefs and chooses a best response to those beliefs as a strategy in this round. Then, after observing their opponents' actions, the players update their beliefs according to some learning rule (e.g. temporal-differencing, Q-learning, or Bayes' rule). The process is then repeated. It is known (Levine, 1998) that if it converges, then the point of convergence is a Nash equilibrium of the game.

4.1 Nash strategies of linear quadratic games

Let us first consider general two-person infinite-horizon simultaneous linear quadratic games

$$x_{k+1} = A_k x_k + B_k^1 u_k^1 + B_k^2 u_k^2 \tag{12}$$

with the cost functions

$$J^1 = \sum_{k=0}^{+\infty} (x_k^T Q^1 x_k + u_k^{1T} R^{11} u_k^1 + u_k^{2T} R^{12} u_k^2) \tag{13}$$

$$J^2 = \sum_{k=0}^{N-1} (x_k^T Q^2 x_k + u_k^{1T} R^{21} u_k^1 + u_k^{2T} R^{22} u_k^2) \tag{14}$$

where $x_k \in R^n$, $u_k^i \in R$, $Q^i > 0$, $R^{ij} > 0$, system (A, Bi) is stabilizable, (A, Ci) is detectable (where $C^{iT} C^i = Q^i$), and $\left| B^{iT} B^i \right| > 0$ for *i*=1, 2, and k=0, 1, 2, …. We assume that both players have perfect information structures, with which both players know the exact system dynamics $x_{k+1} = A_k x_k + B_k^1 u_k^1 + B_k^2 u_k^2$ and measure exact system states x*k*. It is also assumed that the simultaneous game in (12) – (14) has a unique Nash strategy pair for both players. It is well known (Basar & Olsder, 1999) that the Nash strategy pair is specified by

$$u_k^{1*} \triangleq \underset{u_k^1}{\arg\min}\, J^1 = \gamma^1(x_k) = L^1 x_k \tag{15}$$

$$u_k^{2*} \triangleq \underset{u_k^2}{\arg\min}\, J^2 = \gamma^2(x_k) = L^2 x_k \tag{15}$$

where L1 and L2 are defined, respectively, by

$$L^1 = -[R^{11} + B^{1T} K^1 B^1]^{-1} B^{1T} K^1 [A + B^2 L^2] \tag{17}$$

$$L^2 = -[R^{22} + B^{2T} K^2 B^2]^{-1} B^{2T} K^2 [A + B^1 L^1] \tag{18}$$

*K*1 and *K*2 are specified in the following coupled discrete-time algebraic Riccati equations (DAREs) that, by assumption, have a unique positive semi-definite solution.

$$K^1 = Q^1 + L^{1T} R^{11} L^1 + L^{2T} R^{12} L^2 + \left(A + B^1 L^1 + B^2 L^2 \right)^T K^1 \left(A + B^1 L^1 + B^2 L^2 \right) \tag{19}$$

$$K^2 = Q^2 + L^{1T} R^{21} L^1 + L^{2T} R^{22} L^2 + \left(A + B^1 L^1 + B^2 L^2 \right)^T K^2 \left(A + B^1 L^1 + B^2 L^2 \right) \tag{20}$$

4.2 Adaptation schemes

In the one-side adaptation scheme, only one of the two players has perfect information of the cost functions of both players, i.e., the player knows exact *Q*1, *Q*2, *R*11, *R*12, *R*21, *R*22, while the other player only has access to its cost function and does not know the parameters of the cost function of its opponent. WLOG, we assume that player 1 knows exact Q^i , R^{ij} , A, B^i while player 2 has access to Q^2, R^{21}, R^{22}, A, B^i. We also assume that player 1 will apply its state feedback Nash strategies calculated based the information of system dynamics, system states and cost functions of both players. Player 2 knows in advance that its opponent will implement state feedback Nash strategies. Player 2 will estimate the control gain L^1 first, and then calculate/estimate its own control gain L^2.

We assume that player 1, who has perfect information structure, will apply its real state feedback Nash strategies $u^{1*} = L^1x_k$, so we follow the convectional indirect adaptive control design method (Tao, 2003).

Consider the system defined in (12) with fixed controller (15) for player 1, let $\hat{L}^1_k$ be an estimate of the control gain L^1 of player 1, from the point view of player 2. The block diagram of indirect adaptive control system is shown in Fig. 3.

First, we have

$$\hat{x}_{k+1} = Ax_k + B^1\hat{L}^1_k x_k + B^2u^2_k \tag{21}$$

Since matrix $B^{1T}B^1$ is invertible, we obtain, from (14) and (21)

$$[B^{1T}B^1]^{-1}B^{1T}(\hat{x}_{k+1} - x_{k+1}) = (\hat{L}^1_k - L^1)x_k \tag{22}$$

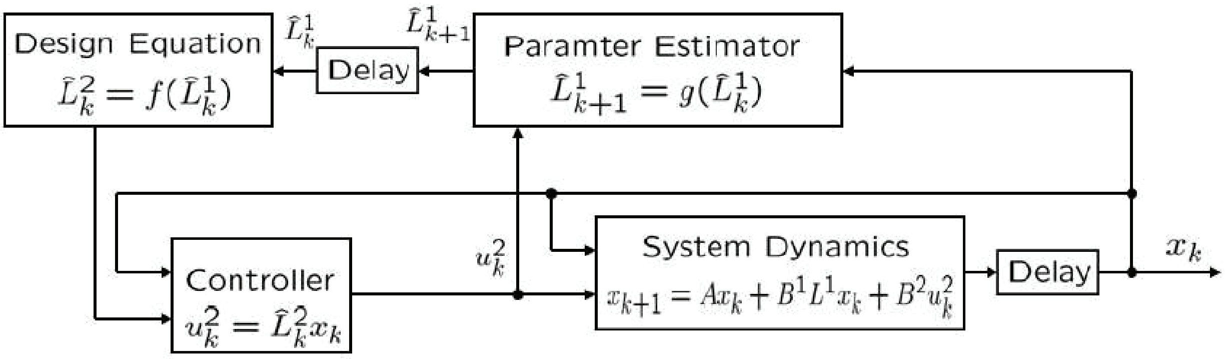

Design Equation
$\hat{L}^2_k = f(\hat{L}^1_k)$
$\hat{L}^1_k$
Delay
$\hat{L}^1_{k+1}$
Paramter Estimator
$\hat{L}^1_{k+1} = g(\hat{L}^1_k)$
u^2_k
Controller
$u^2_k = \hat{L}^2_k x_k$
System Dynamics
$x_{k+1} = Ax_k + B^1L^1x_k + B^2u^2_k$
Delay
x_k

Fig. 3. Indirect adaptive control design for one-side adaptation scheme

We introduce the estimation error $e_k=[B^{1T}B^1]^{-1}B^{1T}(\hat{x}_k - x_k)$, ($e_k$ is a scalar), then

$$e_k = \tilde{L}^1_{k-1}\phi_k \tag{23}$$

where $\tilde{L}^1_{k-1}=\hat{L}^1_{k-1}-L^1$ and $\phi_k=\frac{1}{z}[x]_k=x_{k-1}$. (Note that $\frac{1}{z}[x]_k$ denotes the output of the system with transfer function $\frac{1}{z}$ and input x_k.)

Choose the adaptive law for $\hat{L}^1_k$ as

$$\hat{L}_k^1 = \hat{L}_{k-1}^1 - \frac{e_k \phi_k^T \Gamma}{m_k^2} \tag{24}$$

where $0<\Gamma=\Gamma^T<2I_n$ is a gain matrix, and

$$m_k = \sqrt{\kappa + \phi_k^T \phi_k}, \quad \kappa > 0$$

As shown in Fig. 3, for any $\hat{L}_k^1$, a design function or mapping will be used to calculate $\hat{L}_k^2$. From (18) and (20), we have a best response strategy for player 2 given the estimated control gain $\hat{L}_k^1$ of player 1,

$$\hat{L}_k^2 = -[R^{22} + B^{2T} \hat{K}_k^2 B^2]^{-1} B^{2T} \hat{K}_k^2 [A + B^1 \hat{L}_k^1] \tag{25}$$

$$\hat{K}_k^2 = Q^2 + \hat{L}_k^{1T} R^{21} \hat{L}_k^1 + \hat{L}_k^{2T} R^{22} \hat{L}_k^2 + \left(A + B^1 \hat{L}_k^1 + B^2 \hat{L}_k^2\right)^T \hat{K}_k^2 \left(A + B^1 \hat{L}_k^1 + B^2 \hat{L}_k^2\right) \tag{26}$$

It is proved that the $\hat{K}_k^2$ specified in (25) and (26) is the solution to the DARE

$$\bar{A}^T X \bar{A} - X - \bar{A}^T X \bar{B} \left[\bar{B}^T X \bar{B} + \bar{R}\right]^{-1} \bar{B}^T X \bar{A} + \bar{Q} = 0 \tag{27}$$

where

$$\bar{A}=A+B^1\hat{L}_k^1, \ \bar{B}=B^2, \ \bar{R}=R^{22} \ \text{and} \ \bar{Q}=Q^2+\hat{L}_k^{1T}R^{21}\hat{L}_k^1 .$$

Remark 5: *We can easily obtain* $\hat{K}_K^2$ *by using commercial software such as MATLAB command dare* ($\bar{A},\bar{B},\bar{Q},\bar{R}$).

Similar to the proof of Normalized Gradient algorithm (Tao, 2003, page 115-116), we define a parameter error measurement function $V(\tilde{L}_k^1)=\tilde{L}_k^1\Gamma^{-1}\tilde{L}_k^{1T}$ for the adaptive law in (24). Then,

$$\begin{aligned} V(\tilde{L}_k^1) - V(\tilde{L}_{k-1}^1) &= \left[\tilde{L}_{k-1}^1 - \frac{e_k \phi_k^{\mathrm{T}} \Gamma}{m_k^2}\right] \Gamma^{-1} \left[\tilde{L}_{k-1}^1 - \frac{e_k \phi_k^{\mathrm{T}} \Gamma}{m_k^2}\right]^{\mathrm{T}} \\ &= -\frac{1}{m_k^2} \left(\tilde{L}_{k-1}^1 \phi_k e_k + e_k \phi_k^{\mathrm{T}} \tilde{L}_{k-1}^{1\mathrm{T}} - \frac{\phi_k^{\mathrm{T}} \Gamma \phi_k e_k^2}{m_k^2} \right) \\ &= -\frac{e_k^2}{m_k^2} \left(2 - \frac{\phi_k^{\mathrm{T}} \Gamma \phi_k}{\kappa + \phi_k^{\mathrm{T}} \phi_k} \right) \le -\alpha \frac{e_k^2}{m_k^2} \end{aligned} \tag{28}$$

where $\alpha = 2 - \lambda_{\max}[\Gamma] > 0$. $\lambda_{\max}[\Gamma] \in (0,2)$ is the maximum eigenvalue of Γ which satisfies the condition $0 < \Gamma < 2I_n$. In the above, the equality in the first line comes from (24) and $\tilde{L}_k^1 = \hat{L}_k^1 - L^1$, so

$$\hat{L}_k^1 = \hat{L}_{k-1}^1 - \frac{e_k \phi_k^T \Gamma}{m_k^2} \;\Rightarrow\; \tilde{L}_k^1 = \tilde{L}_{k-1}^1 - \frac{e_k \phi_k^T \Gamma}{m_k^2} \tag{29}$$

Now let's testify the inequality in the last line. By the eigenvalue decomposition of Γ, we can obtain $\Gamma = W^{\mathrm{T}} \Lambda W$ where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the eigenvalue matrix, and W contains the eigenvectors (and W satisfies the condition $W^{\mathrm{T}} = W^{-1}$). Therefore,

$$\phi_k^{\mathrm{T}} \Gamma \phi_k = \phi_k^{\mathrm{T}} W^{\mathrm{T}} \Lambda W \phi_k \le \lambda_{\max}[\Gamma] \phi_k^{\mathrm{T}} W^{\mathrm{T}} W \phi_k = \lambda_{\max}[\Gamma] \phi_k^{\mathrm{T}} \phi_k \tag{30}$$

Since $\kappa > 0$, we have $\kappa + \phi_k^{\mathrm{T}} \phi_k \ge \phi_k^{\mathrm{T}} \phi_k$. Then

$$0 \le \frac{\phi_k^{\mathrm{T}} \Gamma \phi_k}{\kappa + \phi_k^{\mathrm{T}} \phi_k} \le \lambda_{\max}[\Gamma] \tag{31}$$

This completes the verification of (28), from which we have

$$\alpha \sum_{k=1}^{N} \frac{e_k^2}{m_k^2} \le V(\tilde{L}_0^1) - V(\tilde{L}_N^1) \le V(\tilde{L}_0^1) \tag{32}$$

So $\lim\limits_{N \to +\infty} \sum\limits_{k=1}^{N} \frac{e_k^2}{m_k^2}$ is bounded from above by $V(\tilde{L}_0^1)$. Since $m_k^2 = \kappa + \phi_k^{\mathrm{T}} \phi_k > 0$, we obtain

$$\lim_{N \to +\infty} e_k = \lim_{N \to +\infty} \tilde{L}_{k-1}^1 \phi_k = 0 \tag{33}$$

If ϕ_k is persistently exciting, then

$$\lim_{N \to +\infty} \tilde{L}_k^1 = 0 \;\Rightarrow\; \lim_{N \to +\infty} \hat{L}_k^1 = L^1 \tag{34}$$

We proposed two ways to satisfy the excitation conditions (AstrÄom & Wittenmark, 1995, page 63-67). The first one is a reference signal tracking method, and the other is called small system disturbance.

We also attend the adaptation design to two-side adaptation scheme (Fig. 4), in which each player needs to estimate the control gain of its opponent. We assume that each player has access to its own cost function as well as the system dynamics, and does not know the parameters of the cost function of its opponent. i.e., player 1 knows Q^1, R^{12}, R^{11}, A, B^1 and B^2 while player 2 has access to Q^2, R^{21}, R^{22}, A, B^1 and B^2. We also assume that each player knows in advance that its opponent will implement state feedback Nash strategies. So, as shown in Fig. 4, player 2 (or player 1) will estimate the control gain L^1 (or L^2) first, and then calculate/estimate its own control gain L^2 (or L^1). The convergence is proved (Shen, 2006) via two properties of "reaction-curve" like relations between control gains of two players.

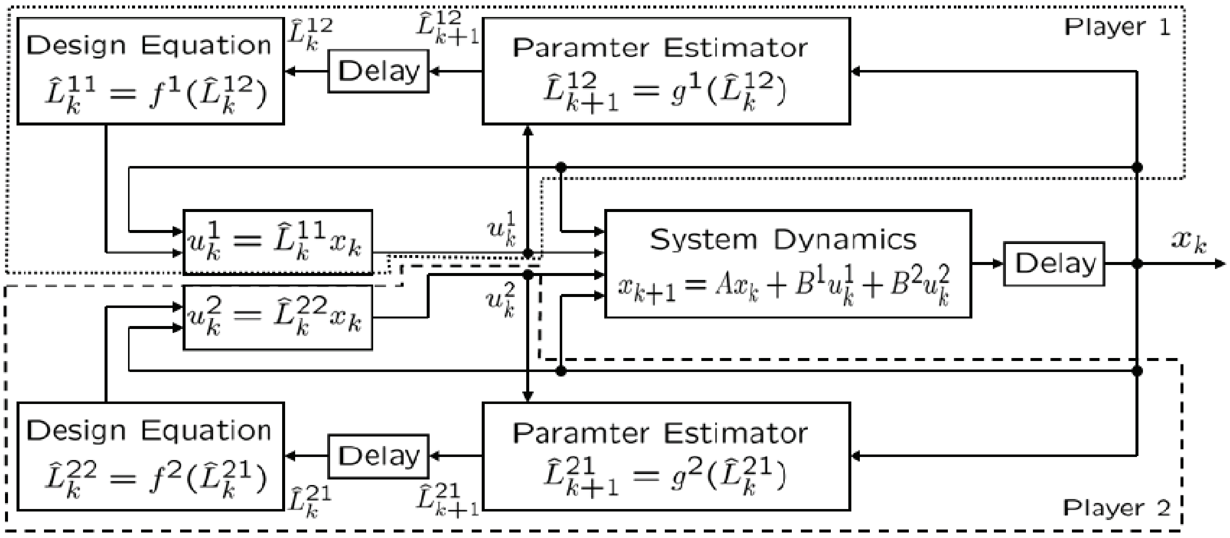

Design Equation
$\hat{L}_k^{11} = f^1(\hat{L}_k^{12})$
$\hat{L}_k^{12}$
Delay
$\hat{L}_{k+1}^{12}$
Paramter Estimator
$\hat{L}_{k+1}^{12} = g^1(\hat{L}_k^{12})$
Player 1
$u_k^1 = \hat{L}_k^{11} x_k$
u_k^1
System Dynamics
$x_{k+1} = Ax_k + B^1u_k^1 + B^2u_k^2$
Delay
x_k
$u_k^2 = \hat{L}_k^{22} x_k$
u_k^2
Design Equation
$\hat{L}_k^{22} = f^2(\hat{L}_k^{21})$
Delay
$\hat{L}_k^{21}$
$\hat{L}_{k+1}^{21}$
Paramter Estimator
$\hat{L}_{k+1}^{21} = g^2(\hat{L}_k^{21})$
Player 2

Fig. 4. Indirect adaptive control design for two-side adaptation scheme

5. Simulation and visualization tool

To evaluate our game theoretic approach for cyber attack prediction and mitigation, we have constructed a Cyber Game Simulation Platform (CGSP) (as shown in Fig. 5) based on an open-source network experiment specification and visualization tool kit (ESVT). Through this event-based, interactive and visual simulation environment, various attack strategies (single stage or multi-staged) and scenarios can be easily played out and the effect of game theoretic attack prediction and mitigation can be visually and quantitatively evaluated.

The implemented network components in this platform includes Computer (host), Switch, Open Shortest Path First (OSPF) Router or Firewall, Link (connection), and (Sub) Network (Simulated by a node).

Besides the ordinary network properties such as processing capacity, bandwidth, Pr{error}, and delay etc., CGSP components can be assigned a number of network attack containment or traffic mitigation properties to act as various defense roles, including smart IDS (intrusion detection systems), incoming traffic block, and outgoing traffic block. Additionally and more importantly, these defense roles or network defense properties can be deployed and re-deployed in real-time during a game simulation run-time based on the local intelligence and orders from higher-level command centers.

The color of a link represents the traffic volume on that link (in KBps and in Mbps). Light Gray: less than 1 percent of bandwidth; Green: more than 1 percent of bandwidth; Yellow: between green and red; Red: more than 30 percent of bandwidth. The color of a host indicates the host status. Red: Infected node; Green: Vulnerable node but not infected; Gray: Non-vulnerable node.

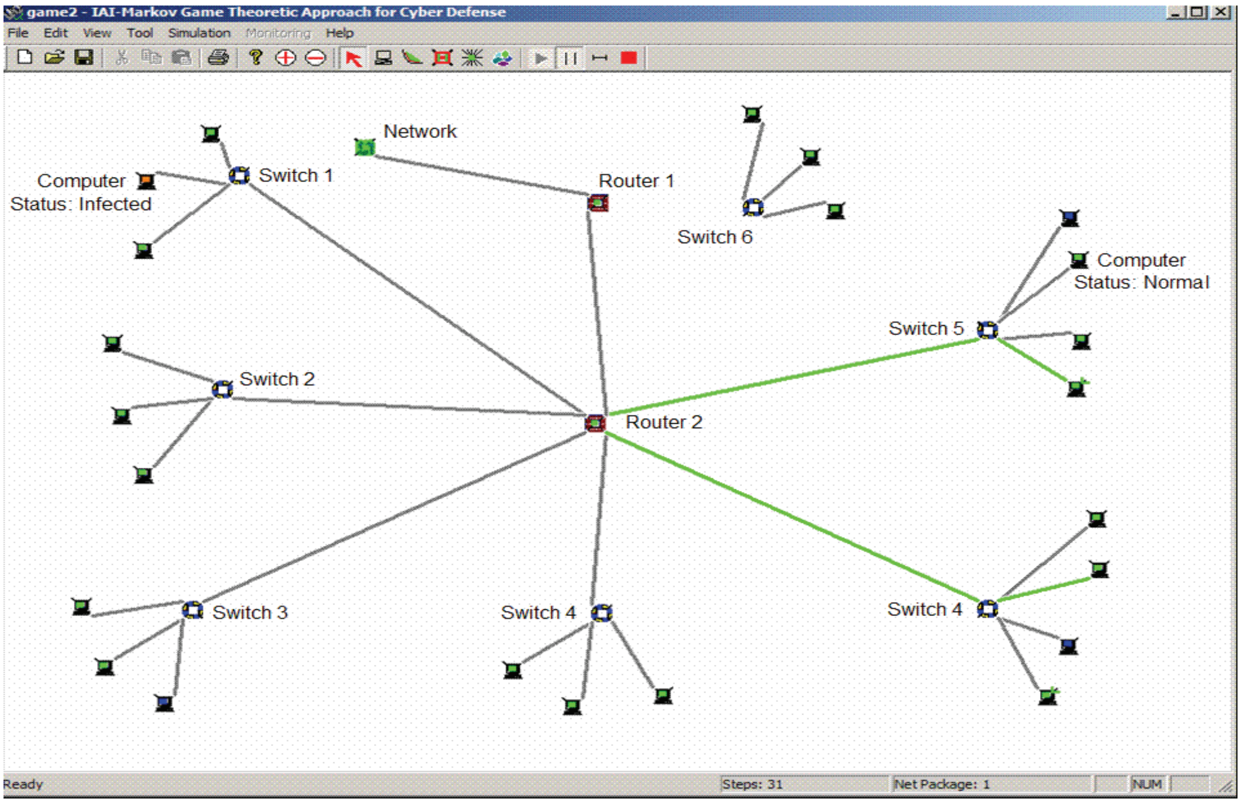


Fig. 5. Cyber Game Simulation Platform (CGSP)

In our simulation platform, network attacks and defenses are simulated in CGSP by events. Live network packets and other communications are represented and simulated by the main network event queue. Users or software agents can inject packets or network events through the timed event (M/M/1) queue. Security alerts or logs are generated and stored in the security log queue.

There are a number of cyber attacks that are included in the CGSP implementation: Port scan, Buffer attack (to gain control), Data bomb or Email bomb from and to a single host, Distributed Denial of service from multiple hosts, Worm attack, and Root right hack (confidentiality loss). [Note: Both buffer attack victims and worm infectives will join the distributed denial of service when they receive the DDOS command.]

The arsenal of network defense team includes: Smart IDS (Accuracy and false positive adjustable), Directional traffic block (outgoing or incoming), Host Shutdown, Host Reset (shutdown and restart). [Note: Both SHUTDOWN and RESET will clear the infection status on the host.]

We simulated a scenario (Fig. 5) with 23 workstations, 2 routers, 7 switches, and 1 network. In this scenario, we first limit the look-ahead steps K to 2 (which means the defense side does not consider the multi-stage attacking patterns).

In this case, we implemented Nash strategies for cyber network defense side. We can see from Fig. 6 and Fig. 7 that a target computer (web server) is infected or hacked. Then the computer (web server) will be used by attacking force to infect other more important target computers such as file servers or email servers. This two-step attacking scheme is based on two facts: 1) a public web server is easy to attack and 2) an infected internal computer (web server in this case) is more efficient and stealthy than an external computer to attack well protected computers such as data servers or email servers.

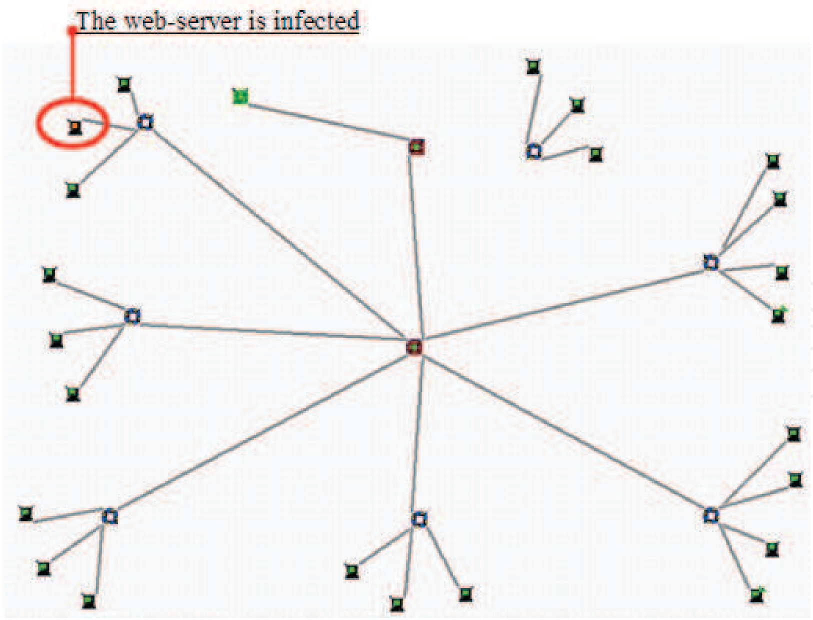


Fig. 6. A public web server is infected or hacked.

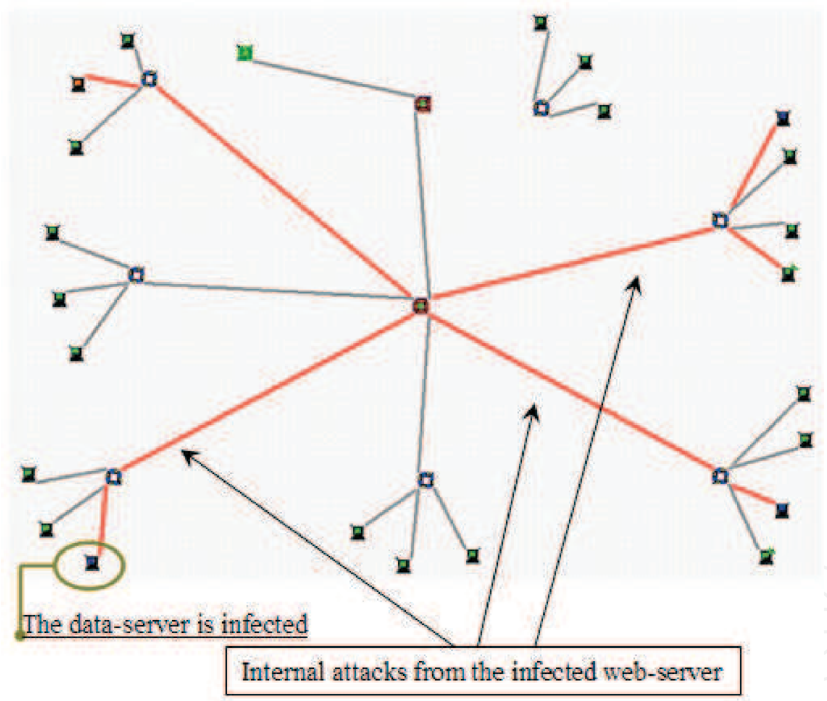

The data-server is infected
Internal attacks from the infected web-server

Fig. 7. Three more important data servers are attacked by the infected internal server.

The adaptive scheme is implemented and the results are shown in Fig. 8. In this plot, $\left\{\hat{L}_k^{ij}\right\}_s$ is the s^{th} element of $\hat{L}_k^{ij}$, which is the estimate of state feedback control gain, *L*, of player *j* by player *i*, (*i*,*j*=1,2). We can see the convergence of $\hat{L}_k^{ij}$ to L_j. which is the actual value of control gain for player *j*. During the adaptation, the overshoots indicate that the decision maker will sometimes overact the changes (deviation from the current Nash equilibria) in the strategies of his opponents.

In the next run, we set the look-ahead step *K*=5. Then no network nodes are infected or hacked during the simulation of 2 hours. If a public server is infected, the defense side can foresee the enemy's next attacking internal server from the infected network node. Then a shut-down or reboot action will be taken to destroy the multi-stage attack at the first stage.

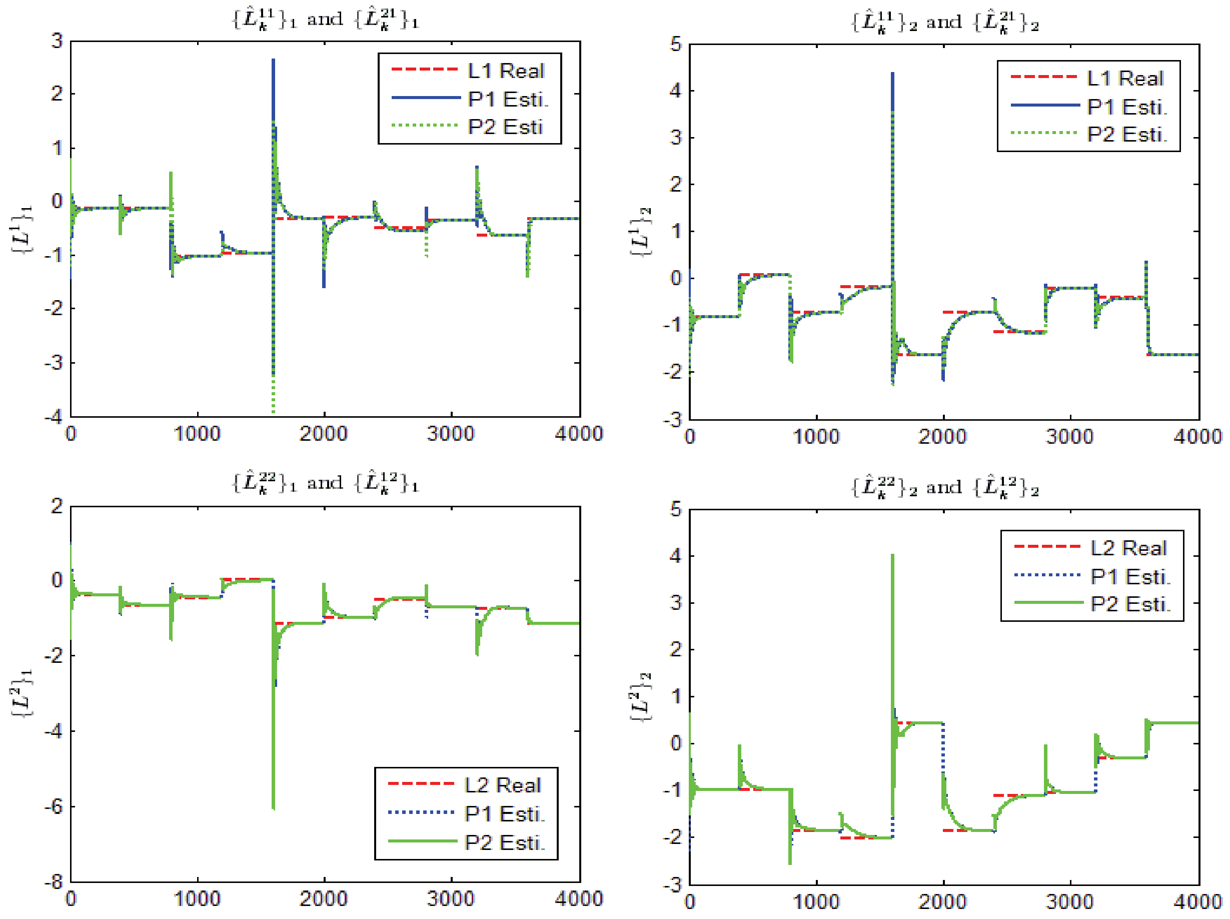


Fig. 8. Results of Adaptive Design

6. Conclusions

We implemented an adaptive Markov game theoretic situation awareness and adversary intent inference approach in a data-fusion/data-mining cyber attack and network defense framework (Fig. 2). The network security system was evaluated and protected from a perspective of data fusion and adaptive control. The goal of our approach was to examine the estimation of network states and projection of attack activities (similar to extended course of action (ECOA) in the warfare scenario). We used Markov game theory's ability to "step ahead" to infer possible adversary attack patterns. With the consideration that the parameters in each game player's cost function is not accessible to other players, we designed an adaptation scheme, based on the concept of Fictitious Play (FP), for the piecewise linearized Markov game model. A software tool was developed to demonstrate the performance of the adaptive game theoretic high level information fusion approach for cyber network defense and simulations were performed to verify and illustrate the benefits of this model.

7. References

Agah, A.; Das, S. K. & Basu, K. (2004). A non-cooperative game approach for intrusion detection in sensor networks, *Vehicular Technology Conference, 2004. VTC2004-Fall.* pp. 2902 – 2906, 2004.

Alpcan, T. & Basar, T. (2003). A game theoretic application to decision and analysis in Network Intrusion Detection, *42nd IEEE CDC 2003*, pp. 2595-2600, Maui, Hawaii, USA, 2003.

AstrÄom, K.J. & Wittenmark, B. (1995). *Adaptive Control*. Addison-Wesley Series in Electrical Engineering: Control Engineering, Addison-Wesley, second ed., 1995.

Basar, T. & Olsder, G. J. (1999). *Dynamic Noncooperative Game Theory*, SIAM Series in Classics in Applied Mathematics, second ed., January, 1999.

Blasch, E. & Plano, S. (2005). DFIG Level 5 (User Refinement) issues supporting Situational Awareness Reasoning, *7th International Conference on Information Fusion*, pp. xxxv-xliii, ISBN: 0-7803-9286-8, Philadelphia, PA, USA, July, 2005.

Brown, G. W. (1951). Iterative solutions of games by fictitious play, In: *Activity Analysis of Production and Allocation* (T. C. Koopmans, ed.), New York: Wiley, 1951.

Chen, G.; Shen, D.; Kwan, C.; Cruz, Jr., J. B.; Kruger, M. & Blasch, E. (2007). Game Theoretic Approach to Threat Prediction and Situation Awareness, *Journal of Advances in Information Fusion*, vol. 2, no. 1, pp. 35-48, June, 2007.

Fudenberg D. & Levine, D. K. (1998). *The Theory of Learning in Games*, Cambridge: MIT Press, 1998.

Nash, J. (1951). Noncooperative games, *Annals of Mathematics*, vol. 54, pp. 286-295, 1951.

Salerno, J.; Hinman, M. & Boulware, D. (2005). A Situation Awareness Model Applied To Multiple Domains, *Proc. Defense and Security Conference*, Orlando, FL, 2005.

Sallhammar, K.; Knapskog, S. J. & Helvik, B. E. (2005). Using Stochastic Game Theory to compute the expected Behavior of attackers, *Proceedings, 2005 Symposium on Applications and the Internet Workshops*, 2005.

Sastry, S. S. (1999). *Nonlinear Systems: Analysis, Stability and Control*, Springer-Verlag, New York, NY, 1999.

Shapley, L. S. (1953). Stochastic games, *Proc. National Academy of Sciences of the United States of America*, vol. 39, pp. 1095-1100, 1953.

Shen, D. (2006). *Nash strategies for dynamic noncooperative linear quadratic sequential games*, Ph.D. Dissertation, Adviser: Jose B. Cruz, Jr., Ohio State University, Electrical Engineering, 2006.

Tadda, G.; Salerno, J.; Boulware, D.; Hinman, M. & Gorton, S. (2006). Realizing Situation Awareness within a Cyber Environment, In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, edited by B. V. Dasarathy, Proc. of SPIE Vol. 6242, SPIE, Bellingham, WA, 2006.

Tao, G. (2003). *Adaptive Control Design and Analysis*, Adaptive and Learning Systems for Signal Processing, Communications and Control Series, Hoboken, N.J.: Wiley-Interscience, 2003.