A Game Theoretic Approach for Attack Prediction *

Peng Liu Lunquan Li
Department of Information Systems, UMBC
Baltimore, MD 21250
{ pliu, lli2 }@umbc.edu

Abstract

The ability to predict attacks can dramatically enhance people's ability to defend cyber attacks since it has the potential to transform existing passive (or reactive) secure systems, where the defender lags behind the attacker, into an active one. Attack prediction can be broken down into two categories: trend prediction and action prediction. In this paper, we present a game theoretic approach for attack prediction, which to our best knowledge is the first approach for action prediction. Our approach models the computer system and the attacker(s) as two self-interested *players* playing a multi-stage *game* where the system wants to maximize its security through its defense operations while the attacker wants to maximize the security loss through his or her attacks. The *Nash equilibria* of the game, which specify the expected-utility maximizing best-response of one player to every other player, indicate valuable action predictions. In addition to predicting attacks, the predictions generated by our approach can also give a good estimation of the maximum possible security loss and tell how the defense should be built. We believe our approach can be used to predict almost every known type of attacks. In particular, we have presented a general game-theoretic attack prediction model for attacks on IDS-protected systems, and a specific prediction model for credit card fraud, and the preliminary simulation results are very encouraging.

Keywords: Attack Prediction, Game Theory, Computer Security

1 Introduction

The ability to predict attacks can dramatically enhance people's capacity to defend cyber attacks since it has the potential to evolve existing passive (or reactive) secure systems into active ones. Either static or adaptive defense (where the system can adapt to system state and environment changes, and the environment involves such factors as attacks and workloads) is *passive*. That is, every defense activity, including dynamic adaptations, is triggered by the effect of attacks. As a result, although when the trend and actions (or patterns) of attacks or workloads do not change or gradually changes, the system can smoothly adapt its behavior to the new environment, when the trend and actions dramatically oscillate, the system can perform very poorly during the adaptation process, and substantial security loss can be caused. *Active* defense takes defensive actions simultaneously with the attacker, so the system no long lags behind the attacker, and the system can adapt to dramatically changed environments in a much smoother way. It is clear that effective active defense needs accurate predictions of attacks. Note that the ability to predict attacks does not imply the ability to prevent attacks.

However, due to the tremendous amount of uncertainty about the attacker's behaviors (e.g., motivations, preferences, actions, the types of attacks), attack prediction is a very challenging task.

*This work is partially supported by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Material Command, USAF, under agreement number F30602-00-2-0575.

Attack prediction has at least the following two aspects: (1) predicting when and where an attack will happen. This aspect may also include such problems as predicting the total number of attacks on a system during a period of time, and predicting the probability that the next system operation is an attack. We call this aspect *trend prediction*. (2) Predicting the actions that the attacker will probably take when an attack happens. We call this aspect *action prediction*. Trend prediction has been studied to some extent in the context of *trend analysis* [BAMF01]. In this paper, we will focus on action prediction, and to our best knowledge, the proposed game-theoretic attack prediction framework is the first approach to action prediction.

Our approach models the computer system and the attacker(s) as two self-interested *players* playing a multi-stage *game* where the system wants to maximize its security through its defense operations while the attacker wants to maximize the security loss through his or her attacks. The *Nash equilibria* of the game, which specify the expected-utility maximizing best-response of one player to every other player, indicate valuable action predictions. The predictions produced by our framework can tell when an attack happens which actions the attacker will probably take, although they cannot tell when the attack will probably happen. The predictions produced by our framework can (1) tell the system which attacking actions can be expected; (2) tell the system which kinds of security loss the system has to tolerate; (3) help the system to estimate the maximum possible amount of security loss that could be caused; (4) tell the system how to configure its security mechanisms in such a way that the total security loss can be minimized.

1.1 Why can game-theoretic attack prediction models be valuable?

We believe game-theoretic attack prediction models can generate valuable (and rational) predictions about attacks for several reasons. First, *noncooperative game theory* is the primary tool to handle *strategic interdependence* (in multiperson decision making and economics) [MCWG95], and strategic interdependence is the fundamental characteristics of the attack-defense relationship in computer security, where each party (the attacker or the defender, namely the secure system) recognizes that (a) the payoff he or she receives (in security gain or loss) depends not only on his or her actions but also on the actions of *other* individuals, and (b) the actions that are best for him or her to take may depend on actions that the other individuals have already taken, on those he or she expects them to be taking at the same time, and even on future actions that they may take, or decide not not to take, as a result of his/her current actions. The presence of strategic interaction in computer security indicates the value of game theory in attack prediction.

Second, game-theoretic models are successfully used to predict *rational* behaviors (or actions) in many applications such as auctions and their *rationality* notion is that each party plays an expected-utility maximizing best-response to every other party. This rationality notion is consistent with the goals of both the attacker (where the utility is measured by the amount of security loss) and the system (where the utility is measured by the amount of security gain). So rational attacker actions predicted by our game-theoretic prediction model can be good attack predictions.

Third, we believe game theoretical models, as reviewed in Appendix A, are a natural mathematical model to analyze the battles between the attacker(s) and the system since: (1) the attacker(s) and the system are "perfect" players of a game where the attacker's strategy or action space is the set of possible attacking actions, and the system's strategy or action space is the set of possible ways to *configure* its security mechanisms. (2) The payoff earned by an attacker during a game play $\{a_{attacker}, a_{system}\}$ indicates when the system's security mechanisms are configured according to a_{system}, to which extend the attacker's action $a_{attacker}$ achieves the attacker's hacking goal. (3) The payoff earned by the system during a game play similarly indicates when the attacker chooses $a_{attacker}$ as the attacking action, to which extend the system's configuration a_{system} achieves the system's security goal. (4) A multi-round battle between the attacker and the system can be easily

modeled by a multi-stage game.

Fourth, we believe the Nash equilibria of attack prediction games can lead to valuable action predictions. In particular, in an attack prediction game, each Nash equilibrium of the game, namely a pair of specific strategies $\{s^*_{attacker}, s^*_{system}\}$ (Note that in multi-stage games a strategy could involve multiple actions), indicates that the best attacking strategy for an rational attacker is $s^*_{attacker}$ when the system chooses s^*_{system} as the defense strategy, and vice versa. Hence, if any one of the two players chooses a Nash equilibrium strategy then for the other player no strategy is better than the other Nash equilibrium strategy. So no player wants to deviate from his or her Nash equilibrium strategy. Therefore, Nash equilibria can lead to *rational* predictions since if the predictions are not a Nash equilibrium, then at least one player will have the incentive to deviate from the prediction about his or her strategy.

However, it should be noticed that game-theoretic models cannot be used to predict *any* attacks. Game-theoretic models assume that (a) the attacker's strategy space is known to the system and vice versa, and (b) the attacker's utility function is known to the system and vice versa. Hence it is clear that game-theoretic models cannot be used to predict *unknown* types of attacks.

1.2 Related Work

Game theory is the study of multiperson decision problems where the payoff of one person's decision is also dependent on the decisions of the other people [Nas50, Har73, HS88, Gib92]. Game theory has revolutionalized almost all fields of economics - industrial, organization, international trade, labor economics, and macroeconomics, to name only a few. In addition, game theory is also used in such non-economic applications as high-speed networking [PSC98], ecosystem management [Vin94], and politics (voting) [Pap94]. The proposed approach, to our best knowledge, is the first application of game theory to attack prediction.

One important application of game theory is *mechanism design* [MCWG95], which has been widely applied in auctions, e-commerce, resource allocation, routing, and survivability. For example, interactive combinational auctions are studied in [WW00, PU00]. A decentralized mechanism design for routing in a multicast tree is proposed in [FPS01]. Decentralized task/resource allocation with no centralized auctioneer is studied in [SL96, San00]. Bidding agents for online auctions are studied in [HRW00]. Multiagent cooperative search for portfolio selection is studied in [PH01]. And market-based network survivability (through optimal decentralized resource allocation after attacks) is studied in [EJK+00]. The goal of mechanism design is to design the "rules of a game", i.e., the actions available to players and the method that is used to compute the outcome based on those actions, in such a way that the designed incentives will encourage player behavior that leads to optimal system-wide solutions despite the self-interest of individual players. Mechanism design is, however, orthogonal to our approach since in our model we assume the rules of a game are pre-determined.

Trend analysis: [BAMF01] shows that the trend of the attacks reported by CERT can be modeled by $C = I + S \times \sqrt{M}$ where C is the cumulative count of reported attacks, M is the time since the start of the attack cycle, and I and S are the regression coefficients determined by analysis of the incident report data.

Intrusion detection (ID) has attracted many researchers [Lun93, MHL94]. The existing methodology of ID can be roughly classed as *anomaly detection* [JV91, JV94, SM97, LX01, SBB01] or *misuse detection* [GL91, IKP95] which is based on known patterns of attacks, called *signatures*, and the idea that an access that matches a signature is an intrusion. In [SFL97], an anomaly credit fraud detection method is proposed and it fits in with our anomaly detection model.

1.3 Our Contributions and Paper Organization

We believe that our game theoretic approach can predict almost every known type of attacks although the predictions it produces for some attacks may not be very valuable. In particular, we have presented a general game-theoretic attack prediction model for attacks on IDS-protected systems, and a specific prediction model for credit card fraud, and the preliminary simulation results are very encouraging. The rest of the paper is organized as follows. In Section 2, we introduce a motivating example. Section 3 presents our game-theoretic attack prediction approach. In Section 4, we present some preliminary game simulation results. Section 5 describes one possible optimization to our basic attack prediction models. In Section 6, we discuss some enforcement issues. Section 7 concludes the paper.

2 A Motivating Example

Using stolen credit cards or numbers is the primary way of credit card fraud, which can cause a credit card company to lose millions of dollars a year. This problem is getting worse in the online world since the Internet supplies perfect environment for thieves to steal credit card and customer information, shop or trade with almost complete anonymity [*]. While cardholders are usually not responsible for fraud charges above $50, it becomes urgent for merchants and credit card companies to find shields and weapons to protect themselves.

How does a credit card transaction get authorized? First the merchant sends the card information to its *acquiring bank*, the bank that provides the merchant with its credit card processing *account*. Through a safe bank network run by some credit card association like Visa and Mastercard, this information is routed to the *issuing bank*, which issues the card. Then the issuing bank performs a number of checks to verify whether the card is valid, whether it is not over limit and has not been reported lost or stolen. If all checks are passed it will send an 'OK' signal to the merchant's acquiring bank, which then issues an authorization to the merchant. The merchant may apply some additional checks, such as signature checks and address verification (in online shopping), before the final authorization of the transaction. During this process, some accounting will be done by these two banks so the transaction money can be transferred from the issuing back's account to the merchant's account (sooner or later).

Beyond these traditional authorization methods some credit card companies and merchants have begun to use modern computer technologies like data mining and intrusion detection to find fraudulent transactions. The most used method is to give every transaction a *suspicion level* according to detection rules prescribed by security experts or "classifiers" [SFL97] generated from data mining approaches (based on transaction history data) to tell if a transaction is suspicious or normal. The assumption of this approach is that either the credit card thieves will commit crimes in a similar way to other thieves whose characteristics have been logged, or that the spending activity of thieves will deviate from the normal behavior of the real cardholder since thieves are typically not able to know the cardholder's shopping history or habit. If the suspicion level of a transaction is above a specific pre-set threshold, the transaction can be rejected. The detection accuracy depends on these rules or classifiers, which usually need to adapt to the latest transactions.

An IDS-integrated credit card authorization system can be as shown in Figure 1. The IDS (Intrusion Detection System) is part of the issuing bank authorization system. For every incoming transaction T_i on card c_j, the IDS computes the suspicion level of T_i, denoted $sl(T_i)$, by measuring how T_i's behavior deviates from the *profile* of c_j, denoted $P(c_j)$, which summarizes the precious shopping behavior associated with c_j. If $sl(T_i)$ is above a specific threshold $TH(c_j)$, the authorization fails, otherwise, the issuing bank sends an 'OK' message to the merchant's acquiring bank. Note that

[*] http://antifraud.com/tips.htm

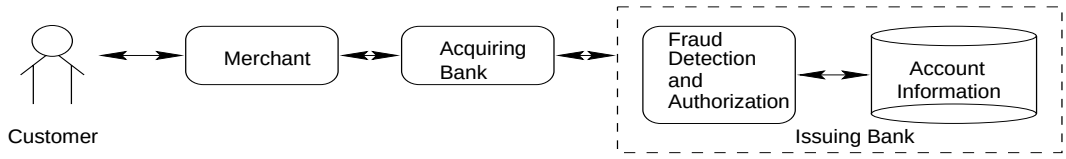

Figure 1: IDS-integrated credit card authorization systems

different cards should have different thresholds. Note also that setting up $TH(c_j)$ faces a tradeoff: if $TH(c_j)$ is too high, many frauds will succeed; if $TH(c_j)$ is too low, serious detail-of-service is possible.

The ability to predict the actions of a fraud credit card transaction (e.g., how much money it will spend) has several merits. First, the predictions can tell the issuing bank which kinds of frauds can be expected. Second, the predictions can tell the issuing bank which kinds of loss the bank has to tolerate. Third, the predictions can help the issuing bank to estimate the maximum fraud loss that could be caused. Fourth, the predictions can tell the issuing bank how to configure the authorization system in such a way that the total fraud loss can be minimized without causing serious denial-of-service.

However, it should be noticed that the ability of our game models to predict frauds does not mean that the predictions produced by our models can be used to prevent frauds or immunize the credit card company from any loss caused by frauds, although the predictions can help reduce the amount of fraud loss. First, action prediction is different from fraud detection and cannot replace detection. Although we can predict what the attacker could do when a fraud happens, we do not know when the fraud will happen. This is the thing that fraud detection tries to do. Second, the predictions generated by our prediction models have the following property, that is, whether or not the attacker will take the predicted action is also dependent on whether or not the fraud detection system will take the predicted defense strategy or configuration (e.g., $TH(c_j)$). This property indicates that in order to make these predictions credible, the fraud detection system cannot deviate from the predicted defense strategy and take a strategy that can defeat the predicted frauds.

3 Our Approach

3.1 The Model

To make our presentation more tangible, we first outline a general formal prediction game model for attacks on IDS-protected systems, then we use the credit card fraud example to show how a practical prediction model can be built.

Definition 1 [Anomaly Detection Systems] An anomaly detection system (ADS) is a 7-tuple $< System, S, H, MU, P, SL, TH >$, where
(1) $System$ is the system that is being attacked,
(2) $S = \{s_1, s_2, ..., s_k, ...\}$ is a set of *subjects* by which the operations on the system are organized,
(3) H is a mapping that maps each subject s_i to a sequence of *operations* done by s_i, called the *history* of s_i,
(4) MU is a mapping that maps each history $H(s_i)$ to a set of sub-sequences; each sub-sequence is called a *monitoring unit*,
(5) P is a mapping that maps each subject s_i to a *profile*, denoted $P(s_i)$,
(6) SL is a mapping that maps each monitoring unit of a subject s_i, denote $mu_j^{s_i}$, as well as $P(s_i)$,

to a *suspicion level* which is either a numerical or linguistic number, and
(7) TH is a set of *thresholds*, denoted $\{th^1, th^2, ..., th^m\}$, where each threshold th^k is associated with a specific alarm (level) and is a mapping that maps each subject s_i to a special suspicion level, such that if $SL(mu_j^{s_i}, P(s_i)) > th^k(s_i)$, then the specific *alarm* associated with th^k will be raised.

Anomaly detection is performed in terms of each subject, i.e., a user, a credit card, or a host. One subject, identified by a subject ID such as a user account, a credit card number, or an IP address, can be associated with many *principals* (e.g., a person that uses a credit card). Some of them could be malicious. $H(s_i)$ indicates the *behavior* of s_i so far. SL can be a very complicated process to compute a synthesized suspicion level where a lot of metrics of operations can be taken into account. Although multiple thresholds can be set to raise different levels of alarms, we assume the cardinality of TH is one for simplicity, we denote this single threshold as th, and we assume that if $SL(mu_j^{s_i}, P(s_i)) > th(s_i)$, then $mu_j^{s_i}$ is reported *malicious*, otherwise, it is believed *legitimate*.

To illustrate, consider the credit card fraud detection system shown in Figure 1. The attacker attacks the credit card system by issuing fraudulent credit card *transactions*. S is the set of credit cards. (Note that each customer can have several credit cards, and the shopping behaviors of the same customer on different cards could be very different.) Each operation taken by a credit card c_i is a transaction. And the sequence of transactions taken by a card is the history of the card. Each monitoring unit of a card c_i is a transaction by c_i, denoted $T_j^{c_i}$. Each credit card transaction $T_j^{c_i}$ can be characterized by a set of *attributes*, denoted $\{a_1^{c_i}, a_2^{c_i}, ..., a_n^{c_i}\}$, for example, the amount of transaction money, the merchant and its characteristics, when and where the transaction is issued, the total number of items bought, the types of the items bought, etc.. The shopping characteristics of the transactions by c_i indicate the shopping habit of c_i when there are no frauds. This shopping habit is specified by the profile of c_i, denoted $P(c_i)$. The rational of this fraud detection system is that since the attacker usually does not know the shopping habit of c_i, it is very possible that the attacker uses c_i in a very *abnormal* way. $SL(T_j^{c_i}, P(c_i))$, the suspicion level of a transaction $T_j^{c_i}$, indicate the extent to which the shopping behavior of $T_j^{c_i}$ deviates from the habit of c_i. Note that $SL(T_j^{c_i})$ compares the difference between $SL(T_j^{c_i})$ and $P(c_i)$ from many aspects. $th(c_i)$ indicates the maximum possible amount of anomaly of a legitimate transaction by c_i.

We take a game theoretic approach to predict what the attacker will do to attack a computer system protected by an ADS. In this section, we assume the ADS can report the suspicion level of a monitoring unit before the operations of the monitoring unit are done. (Although in many systems the ADS raises alarms after monitoring units are done, and although predicting the attacks on such systems is beyond the scope of this paper, we have found that the game models presented here can be extended to predict the actions (or effects) of the successful attacks on such systems from a survivability perspective.) In particular, we model the users (or subjects) and the system as two players playing a game. Since the user could be either a *good* guy or a *bad* guy (i.e., attacker), we use Bayesian games which can model multiple types of users to build our model. A simple review of game theory is given in Appendix A.

Definition 2 [Attack Prediction Game] An attack prediction game APG is a specific 2-*player* finitely repeated Bayesian game between the system (i.e., the ADS) and a subject. It is denoted by
$APG = \{A_{ads}, A_{subject}; T_{ads}, T_{subject}; p_{ads}, p_{subject}; u_{ads}, u_{subject}\}$ where
(1) A_{ads} is the *action* space for the ADS, and $A_{subject}$ is the action space for the subject. Each action of the subject is a monitoring unit. Each action of the ADS is an instance of
$\{th^1(subject), th^2(subject), ..., th^m(subject)\}$.
(2) The game has a finite number of *plays* (or stages) and each play includes a pair of simultaneous actions $(a_{subject}, a_{ads})$.
(3) The *type* space for the ADS is $T_{ads} = \{t^{ads}\}$, and the type space for the subject is $T_{subject} = \{t_1^g, t_2^b, t_3^b, ..., t_n^b\}$.

(4) p_{ads} is player ADS's *belief*. $p_{ADS}(t_i^{subject}|t^{ads})$ describes ADS's uncertainty about the subject's possible types, given ADS's own type t^{ads}. Similarly, $p_{subject}$ is determined by $p_{ADS}(t^{ads}|t_i^{subject})$ (for every type of the subject).
(5) u_{ads} is the *payoff* function of the ADS; $u_{subject}$ is the payoff function of the subject. Each player's *type*, privately known by the player, determines the player's payoff function. Here (in each game play) we have n payoff functions for the subject, each one is denoted by $u(a_{ads}, a_{subject}; t_i^{subject})$. And we have one payoff function for the ADS, denoted by $u(a_{ads}, a_{subject}; t^{ads})$.

Remarks. First, note that there could be multiple types of the subject. Here we assume that the subject is associated with one good principal, i.e., t_1^g, and possibly several bad principals. The good principal indicates the legitimate representative of the subject. The bad principals indicate different attackers. For example, a credit card can be fraudulently used by multiple bad guys during the same period of time. Second, types determine payoff functions. It is clear that if the current principal is a bad one, then the ADS want to reject the subject's actions. If the principal is a good one, rejecting the subject's actions will cause denial-of-service.

Third, note that this game is with *incomplete* information, that is, during each play the ADS is uncertain about the subject's payoff function because the ADS is uncertain about the subject's type. If the ADS knows whether or not the current principal is good or bad, intrusion detection is not necessary. We model this uncertainty by probabilities. In particular, in each play, $p_{subject}(t^{ads}|t_i^{subject}) = 1.0$, and $\sum_{i=1}^{n} p_{ADS}(t_i^{subject}|t^{ads}) = 1$. Here we assume $p_{ADS}(t_i^{subject}|t^{ads})$ (for each i) are pre-known.

Fourth, the game can have multiple stages and each stage involves a play. A later stage can repeat a previous stage. And a later stage can be affected by the results of a previous stage. Fortunately, according to Proposition 1 in Appendix A, we know that the Nash equilibria of the game in each play compose the *subgame-perfect outcome*, or the overall outcome, of this multi-stage repeated attack prediction game.

An example APG model for credit card fraud can be as follows. For simplicity, we assume the ADS has only one threshold, i.e., $th(c_i)$. We assume each credit card transaction $T_j^{c_i}$ is characterized by only one attribute, i.e., $amount(T_j^{c_i})$. We assume that $th(c_i)$ is just a number, and correspondingly we assume that profile $P(c_i)$ is also a number. We denote the credit limit of c_i as $CL(c_i)$. We call $[max(0, P(c_i) - th(c_i)), min(CL(c_i), P(c_i) + th(c_i))]$ the *acceptance window* of the ADS because only the transactions within the window can be authorized. We denote the size of this window as $wsize = min(CL(c_i), P(c_i) + th(c_i)) - max(0, P(c_i) - th(c_i))$.

[An APG for credit card fraud prediction]
(1) The game has two players: a subject c_i and the ADS.
(2) A_{ads} is the set of possible values for threshold $th(c_i)$. These values can be either continuous or discrete.
(3) A_{c_i}, the set of possible credit card transactions, is specified by the values of $amount(T_j^{c_i})$ for each transaction $T_j^{c_i}$.
(4) $T_{ads} = \{t^{ads}\}$; $T_{subject} = \{good, bad\}$. We assume at one point of time, there is only one fraud on c_i.
(5) We assume $p_{ADS}(good|t^{ads}) = 1 - \theta$, and $p_{ADS}(bad|t^{ads}) = \theta$.
(6) $u_{subject}(th(c_i), T_j^{c_i}; good) = \begin{cases} 0 & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| \le th(c_i) \\ -DoS(T_j^{c_i}) & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| > th(c_i) \end{cases}$
(7) $u_{subject}(th(c_i), T_j^{c_i}; bad) = \begin{cases} amount(T_j^{c_i}) & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| \le th(c_i) \\ 0 & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| > th(c_i) \end{cases}$
(8) $u_{ads} = (1 - \theta)u_{ads}^{good}(th(c_i), T_j^{c_i}; t^{ads}) + \theta u_{ads}^{bad}(th(c_i), T_j^{c_i}; t^{ads})$, where

$$u_{ads}^{good}(th(c_i), T_j^{c_i}; t^{ads}) = \begin{cases} b.wsize & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| \leq th(c_i) \\ 0 & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| > th(c_i) \end{cases}$$

$$u_{ads}^{bad}(th(c_i), T_j^{c_i}; t^{ads}) = \begin{cases} -amount(T_j^{c_i}) & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| \leq th(c_i) \\ 0 & \texttt{if } |amount(T_j^{c_i}) - P(c_i)| > th(c_i) \end{cases}$$

Remarks. First, note that the investigation cost associated with the credit card company either when a good customer suffers from denial-of-service, or when a fraud is reported, is not taken into consideration because (1) no matter a fraud transaction is detected or not, the investigation is typically unavoidable; (2) if we do not consider the investigation for bad guys, then there is no need to consider the investigation for good guys; (3) all the investigations are performed after the game is over.

Second, note that here $SL(T_j^{c_i}, P(c_i)) = |amount(T_j^{c_i}) - P(c_i)|$. This is a reasonable mapping when $th(c_i)$ is much smaller than $CL(c_i)$. Some other SL mappings can also be valuable, for example, uni-lateral matching where only if $amount(T_j^{c_i}) > P(c_i) + th(c_i)$ will $T_j^{c_i}$ be reported malicious. Third, $u_{subject}(th, T_j^{c_i}; good)$ means that when the customer is good, if $T_j^{c_i}$ is authorized, he or she loses nothing, otherwise, he or she will suffer some denial-of-service loss, which is measured by function $DoS(T_j^{c_i})$. $DoS(T_j^{c_i})$ can be affected by many factors, which are out of the scope of this paper.

Fourth, $u_{subject}(th, T_j^{c_i}; bad)$ means that for a bad guy, if $T_j^{c_i}$ is authorized, then he or she wins $amount(T_j^{c_i})$, otherwise, he loses nothing. Although there is some risk that he or she could be captured by the police, since the risk should be a part of both of his or her payoffs no matter $T_j^{c_i}$ is authorized or not, we remove the risk part from both payoffs for simplicity without losing soundness.

Fifth, u_{ads} indicates the *expected* payoff of the ADS. In particular, u_{ads}^{good} means that if the good guy's transaction can go through, the ADS wins some service availability, which is measured by $b.wsize$ where b is called the *availability weight*, and $wsize$ is called the *availability bandwidth*. $b.wsize$ encourages the ADS to set up a larger acceptance window. u_{ads}^{bad} means that if a transaction of the bad guy is authorized, the IDS loses $amount(T_j^{c_i})$, otherwise, losing nothing.

Sixth, the game can involve multiple plays, although only one play is specified. Although after a game play, (a) the available balance of the card, namely $CL(c_i)$, will be reduced by $amount(T)$, and (b) the profile $P(c_i)$ may need be adjusted, the next play can be modeled in the same way. And according to Proposition 1 in Appendix A, the Nash equilibrium of each play also should compose the subgame-perfect outcome of the whole game.

3.2 A Naive Approach to Prediction

Assume that profile $P(c_i)$ is known to both the good guy and the bad guy. For each play of the game, if the play has a Nash equilibrium $(th^*(c_i), A_g^*, A_b^*)$ where A_g^* and A_b^* denote the Nash equilibrium actions of the good guy and the bad guy, respectively, it much satisfy:

$$|A_g^* - P(c_i)| \leq th^*(c_i) \quad (1)$$

$$A_b^* = P(c_i) + th^*(c_i) \quad (2)$$

$$th^*(c_i) \text{ solves } \max_{th(c_i)} \ (1-\theta)b.wsize - \theta.A_b^* \quad (3)$$

It is clear that when $b = 0$, the play has the following pure-strategy Nash equilibrium: $(0, P(c_i), P(c_i))$, that is, the ADS will choose 0 as the value of $th(c_i)$, and both the good guy and the bad guy will choose $P(c_i)$ as the transaction amount. $(0, P(c_i), P(c_i))$ satisfies requirements (1), (2), and (3), and maximizes the payoffs for each (type of) player given the actions taken by the other players. $(0, P(c_i), P(c_i))$ is a reasonable prediction based on the assumption that $P(c_i)$ is known to the attacker. Although the attacker will always win $P(c_i)$ according to the prediction, if the ADS does not choose the predicted action, the attacker can win more and the ADS can lose more. On the other hand, if the bad guy wants to win more and spends more than $P(c_i)$, then the bad buy's transaction will be rejected if the ADS chooses the predicted action.

However, when $b > 0$ there is no pure-strategy Nash equilibrium since (1) $(0, P(c_i), P(c_i))$ is not a Nash equilibrium because when the attacker chooses $P(c_i)$, the ADS wants to make $th(c_i)$ as large as possible to gain more service availability, but when the ADS chooses $th(c_i) > 0$, the attacker will not stay with $P(c_i)$ and will choose $P(c_i) + th(c_i)$, and (2) every player wants to and can outguess the other on matter which action is taken by the other player. In any game if each player would like to outguess the other(s), there is no pure-strategy Nash equilibrium because the solution to such a game necessarily involves uncertainty about what the other player(s) will do.

3.3 A Probabilistic Approach

The reason that the attacker will always win $P(c_i)$ in the naive approach is because the naive approach assumes that the attacker knows $P(c_i)$. However, this assumption is not practical in general because it is very difficult, if not impossible, for the attacker to exactly know the shopping habit of the good guy.

To produce more practical predictions, in this section we assume that only a probabilistic *distribution* of $P(c_i)$ is known to the attacker. Based on the public statistics of customers' shopping behaviors, the attacker should be able to know the distribution. The distribution can be specified by a *density* function $f(x)$, where x is defined on $[0, CL(c_i)]$. Value $f(x)$ is the probability that the value of $P(c_i)$ is x.

For each play of the game, if the play has a pure strategy Nash equilibrium $(th^*(c_i), A_g^*, A_b^*)$, then first A_g^* will solve:

$$\max_{A_g} \ u_{subject}(th^*(c_i), A_g; good) \tag{4}$$

It is easy to see that the result of (4) is

$$A_g^* = \{A_g \mid \ |A_g - P(c_i)| \leq th^*(c_i)\} \tag{5}$$

Second, A_b^* will solve:

$$\begin{aligned} &\max_{A_b} \ u_{subject}(th^*(c_i), A_b; bad) \\ &= \max_{A_b} \ A_b.prob(|A_b - P(c_i)| \leq th^*(c_i)) \end{aligned} \tag{6}$$

Since the probability that $|A_b - P(c_i)| \leq th^*(c_i)$ is equal to the probability that $P(c_i)$ is within $[max(0, A_b - th^*(c_i)), min(CL(c_i), A_b + th^*(c_i))]$. (We simply denote the window as $[r_1, r_2]$.) Formula (6) is equivalent to

$$\max_{A_b} \ A_b . \int_{r_1}^{r_2} f(x)dx \tag{7}$$

We found that there usually exist first-order conditions in (7). Note that since the attacker does not know $P(c_i)$, it can be the attacker's best strategy to maximize the expected payoff based on a distribution of $P(c_i)$, as shown in (6) and (7). Finally, $th^*(c_i)$ will solve:

$$\max_{th(c_i)} \ (1-\theta)b.wsize - \theta.A_b^*.h(A_b^*, P(c_i), th(c_i)) \tag{8}$$

where

$$h(u, v, w) = \begin{cases} 1 & \texttt{if} \ \ |u - v| \leq w \\ 0 & \texttt{if} \ \ |u - v| > w \end{cases}$$

When $b = 0$, to make (8) maximum, we can make $h(A_b^*, P(c_i), th(c_i)) = 0$, which implies that $th^*(c_i) < |A_b^* - P(c_i)|$, or the limitation of $th^*(c_i)$ should be $|A_b^* - P(c_i)|$. When $b \neq 0$, if b is fairly small, to make (8) maximum, we should have

$$th^*(c_i) \approx |A_b^* - P(c_i)| \tag{9}$$

Based on (5), (9), and the first conditions of (7), we can get a Nash equilibrium $(th^*(c_i), A_b^*, A_g^*)$.

3.4 Adding More Uncertainty

Every Nash equilibrium of the plays specified in the previous section says that when b is fairly small $th^*(c_i) \approx A_b^* - P(c_i)$, that is, when the ADS would like to sacrifice the service availability, every fraud transaction of the attacker will be rejected except when $A_b^* = P(c_i)$ if the attacker follows the predictions (Note that the probability that $A_b^* = P(c_i)$ is almost zero). The main reason that makes these predictions not very meaningful or practical is because the ADS has more knowledge than the attacker. The ADS has no uncertainty about the attacker's actions but the attacker has substantial uncertainty about the ADS's actions. In particular, the ADS knows $P(c_i)$ and $f(x)$, and knows that the bad guy chooses his or her actions based on $f(x)$, however, the attacker does not know $P(c_i)$. As a result, based on the prediction model, the ADS can compute both A_b^* and $th^*(c_i)$, but the attacker can know neither of them (because $P(c_i)$ is necessary to compute A_b^* and $th^*(c_i)$).

In this section, we want to make the attack prediction game more practical. In particular, we want to increase the ADS's uncertainty about the attacker's actions. The rational is that if the attacker knows that if he or she follow the predictions made in the previous section he or she may never succeed (when b is small), the attacker will not take the predicted action and the attacker should know that increasing the randomness of his or her actions can increase the number of successful frauds.

In our model, we assume the attacker's action ($T_j^{c_i}$) is *randomly* chosen from a continuous *range* $[y, y+B]$, which is a part of $[0, CL(c_i)]$. Here we call B the *attack range*. We assume the distribution of $amount(T_j^{c_i})$ over this range, denoted $g(z)$, and B are known to both the ADS[†]. For example, $g(z)$ could be an uniform distribution. The attacker's action is actually specified by y. Note that the prediction model proposed in the previous section is a special case of this game, i.e., when $B = 0$.

If a play of this game has a Nash equilibrium $(th^*(c_i), A_g^*, y^*)$, first it must satisfy:

$$|A_g^* - P(c_i)| \leq th^*(c_i) \tag{10}$$

Second, y^* will solve

$$\max_y \ \int_y^{y+B}(z.g(z).\int_{r_1}^{r_2} f(x)dx)dz \tag{11}$$

Note that here what the ADS wants to do is to maximize the expected payoff based on a distribution of A_b. Third, $th^*(c_i)$ will solve

$$\max_{th(c_i)} \ (1-\theta)b.wsize - \theta \int_{y^*}^{y^*+B} z.g(z).h(z, P(c_i), th(c_i))dz \tag{12}$$

As a result, both $th^*(c_i)$ and y^* are a function of $f(x)$, $g(z)$, $P(c_i)$, $CL(c_i)$, θ, b, and B. The ADS knows all of them (so the ADS knows y^*), but the attacker does not know $P(c_i)$. Nevertheless, in this case, the attacker's fraud transaction will not always be rejected if the attacker takes y^*. To illustrate, first, if $B \geq CL(c_i)$, then it is clear that $y^* = 0$. So there exists a non-zero probability that the attacker can succeed no matter what $th(c_i)$ is. For example, even if $th(c_i) = 0$, there is a non-zero probability that $A_b = P(c_i)$.

Second, if $B < CL(c_i)$, then it is possible that $P(c_i)$ is less than y^* and not in $[y^*, y^*+B]$. In this case, the attacker will win nothing for a set of $th(c_i)$. In particular, when b is very small, $th^*(c_i) \approx y^* - P(c_i)$. However, when b is not very small, it may be worthy to trade vulnerability (or possible fraud loss) for service availability. Then the acceptance window $[max(0, P(c_i) - th^*(c_i)), min(P(c_i) + th^*(c_i), CL(c_i))]$ and the attack range $[y^*, y^*+B]$ may still partially overlap. Third, if $P(c_i)$ is within $[y^*, y^*+B]$, then no matter what $th(c_i)$ is, there is a non-zero probability that the attacker can succeed. In this case, when the ADS chooses a $th(c_i)$ smaller than $th^*(c_i)$, the ADS may lose a lot in service availability with a little gain in reducing the vulnerability, based on the way the ADS

[†]More uncertainty can be added by making B or $g(z)$ uncertain to the ADS.

tradeoffs between availability and vulnerability, which is determined by the value of b. Fourth, when $P(c_i) > y^* + B$, the situation is similar to case two.

Finally, since taking y^* can maximize the expected payoff of the attacker when the ADS takes $th^*(c_i)$, the attacker should be willing to take y^* as his or her action. However, the attacker does not know exactly what y^* is because the attacker does not know $P(c_i)$ (Note that here we assume that the attacker knows b). As a result, the attacker may only be able to get an estimation of y^* and earn less payoff. y^* can be estimated in several ways. For example, based on the simulation results (for every $P(c_i)$) presented in Section 4 which the attacker is able to compute, the attacker can know that when $B = 20$, $b = 0.001$, a value between \$45 and \$50 is a pretty good estimation of y^* with accuracy probability $\int_{27}^{93} f(x)dx$. However, it should be noticed that in many cases (e.g., when $B = 40$, $b = 0.06$) y^* may not be able to be accurately estimated because in these cases if y^* could be accurately estimated, then the attacker can use the estimated y^* to accurately estimate $P(c_i)$, however, if the attacker can accurately estimate $P(c_i)$, then the rationality assumption that the attacker will maximize his or her expected utility will no longer be valid. Although the fact that y^* may not be able to be accurately estimated in some cases will decrease the accuracy of the predictions produced, we believe that the decreased accuracy is, to a large extent, due to incomplete knowledge and information asymmetries, which are the inherent characteristics of the attacker-defender relationship. This fact, to some extent, tells why attack prediction is difficult.

3.5 Predicting Simultaneous Attacks from Multiple Attackers

Although till now the presentation focuses on how to predict the attacks by a single attacker, our attack prediction games can be easily extended to predict simultaneous attacks by multiple attackers. Taking credit card fraud as an example, there are two categories of simultaneous attacks: (1) there are multiple credit cards and multiple frauds, however, for each card there are at most one owner and at most one attacker. (2) There are multiple owners and multiple attackers on one card.

Our attack prediction games can be easily extended to cover these two categories. In particular, for category 1 simultaneous attacks, one game can be associated with each card during the card's life cycle, and the games associated with multiple cards are independent of each other. Hence, the prediction games we have proposed for a single card can be directly applied. For category 2 simultaneous attacks, type space $T_{subject}$ can be extended in such a way that each co-owner of the card belongs to a separate type and each attacker belongs to a separate type. Then the attack prediction games we have proposed, which handle multiple subject types by nature, can be applied to predict simultaneous attacks. In summary, our attack prediction games are by nature multi-player games that can predict simultaneous attack from multiple attackers.

4 Preliminary Results and Analysis

We have done intensive computer simulations on the game plays specified in the previous section. And some preliminary results are shown in Figures 2-9, where $CL(c_i) = 100$, $\theta = 0.05$ (We found the value of θ has no big impact on the results), $f(x)$ is a $(50, 1)$ normal distribution, and $g(z)$ is an uniform distribution. We simulate the game plays with different value combinations of B, b, and $P(c_i)$. Note that the results are associated with a single game stage, and for repeated games these results are enough to represent the predictions made by the whole game.

In general, a preliminary analysis of the results shows that (1) there exist pure strategy Nash equilibria, and there typically exists a single Nash equilibrium for each game play. For example, when $B = 20$, $P(c_i) = 30$, $b = 0.001$, the Nash equilibrium or the prediction is $(y^* \approx 48; th(c_i)^* \approx 18)$, that is, the ADS should have an acceptance window with size 36, and can expect frauds with \$48 transaction amount. (2) As the availability weight b increases, the predicted $th(c_i)^*$ will increase,

and the predicted attack action $y*$ will also increase. This indicates that the ADS would like to sacrifice some fraud loss to increase service availability. (3) The attacker success rate, i.e., the ratio of the fraud transactions that succeed to all the fraud transactions, is usually not zero (See Figures 6 and 7). In particular, as b increases, the attacker success rate will also increase for the same reason as (2). (4) As b decreases, both the ADS's payoff and the attacker's payoff will decrease since the ADS's acceptance window will shrink and ADS's service availability will decrease. (5) Increasing B, or the ADS's uncertainty about the attacker's behavior, increases the values of the attacker's Nash

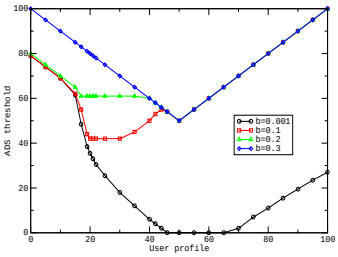

Figure 2: Nash equilibrium ADS strategies for each $P(c_i)$ when B=20 and $\theta = 0.05$

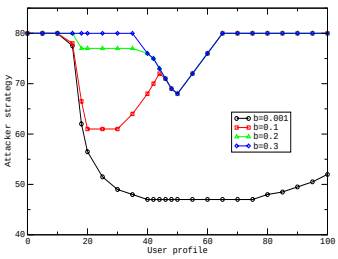

Figure 3: Nash equilibrium attacker strategies for each $P(c_i)$ when B=20 and $\theta = 0.05$

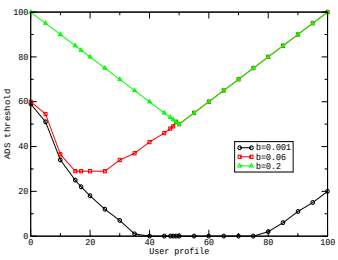

Figure 4: Nash equilibrium ADS strategies for each $P(c_i)$ when B=40 and $\theta = 0.05$

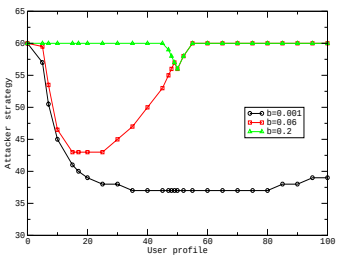

Figure 5: Nash equilibrium attacker strategies for each $P(c_i)$ when B=40 and $\theta = 0.05$

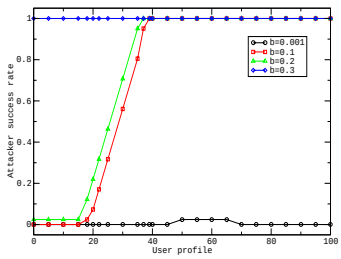

Figure 6: Attacker success rates for each $P(c_i)$ when B=20 and $\theta = 0.05$

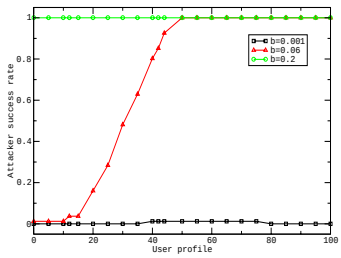

Figure 7: Attacker success rates for each $P(c_i)$ when B=40 and $\theta = 0.05$

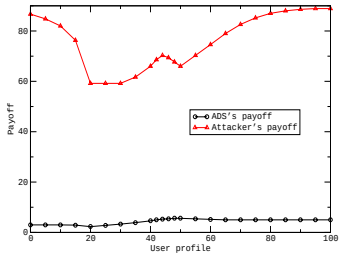

Figure 8: Player payoffs for each $P(c_i)$ when $B = 20$, $b = 0.1$, and $\theta = 0.05$

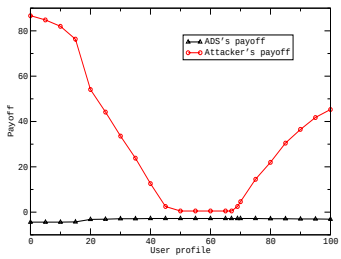

Figure 9: Player payoffs for each $P(c_i)$ when $B = 20$, $b = 0.001$, and $\theta = 0.05$

equilibrium strategies.

The four major merits of our action prediction approach mentioned in Section 1 can be easily justified by these preliminary results. In particular, the Nash equilibria indicate which kind of frauds can be expected and how should the bank configure its ADS, and the corresponding payoffs of these Nash equilibria indicate the maximum amount of fraud loss that the bank has to tolerate.

We have found some interesting implications of these results. First, the ADS should determine the value of the availability weight (or b) based on the profile. From Figures 8 and 9, we can see that when $P(c_i)$ is around \$55, choosing a smaller b can significantly reduce the payoff of the attacker without significantly reducing the ADS's payoff. And when $P(c_i)$ is very small, choosing a larger b can significantly increase the payoff of the ADS without significantly increasing the attacker's payoff. Note that this good feature is due to the fact that in our model the attacker's uncertainty about the ADS is more than the ADS's uncertainty about the attacker. Decreasing the amount of information asymmetries between the ADS and the attacker can reduce the advantages of the ADS.

Second, the results show that (a) the cost of availability is higher attacker success rates and more security loss (i.e., attacker payoffs), and (b) the existence of Nash equilibria is due to the fact that the ADS needs to tradeoff between the goal of minimum fraud loss and goal of maximum service availability. The oscillations of the ADS's Nash equilibrium strategies in Figure 2 give a fine explanation of this tradeoff. In particular, when $b = 0.3$, the availability gain dominates the ADS's payoff, so the ADS thresholds always make the size of the acceptance window $CL(c_i)$ (i.e., the largest). When $b = 0.001$, the fraud loss dominates, so the thresholds try to minimize the attacker's success rate (as shown in Figures 6 and 7). Especially when $P(c_i)$ is smaller than \$45 or larger than \$65, the thresholds yield no overlap between the attack range and the acceptance window. When $b = 0.1$, neither availability nor fraud loss dominates (for every $P(c_i)$). In particular, (a) when $P(c_i)$ is less than \$15, the ADS can have substantial availability without causing a lot fraud loss, so the thresholds are large; (b) when $P(c_i)$ is between \$15 and \$19, the thresholds sharply go down because if not the corresponding availability gain cannot afford the corresponding extra fraud loss; (c) when $P(c_i)$ is between \$20 and \$30, the thresholds do not decrease because if they decrease the corresponding fraud saving cannot afford the availability loss; (d) when $P(c_i)$ is between \$30 and \$45, the thresholds increase because in this way although y^* will also increase, the overlop between the attack range and the acceptance window will not change. Otherwise, if thresholds decrease or do not change, the overlap will increase. Note that here the overlap is the main factor for the fraud loss. (e) when $P(c_i)$ is larger than \$45, the availability dominates.

Third, it should be noticed that the payoffs shown in Figures 8 and 9 are the expected payoffs of the attacker (ADS) instead of his or her *real* payoffs. When b is very small and when $P(c_i)$ is very small or very large, the attacker's expected payoff (as shown in Figure 9) is actually much larger than his or her real payoff, which is actually around zero, although when $P(c_i)$ is round the medium the expected payoff and the real payoff of the attacker are pretty similar. The difference between expected and real payoffs indicates the impact of incomplete knowledge and information asymmetries.

Fourth, the predictions generated have interesting implications on the false alarm rate and the detection rate. In particular, (a) the false alarm rate is dependent on the behavior of the good guy. If the good guy always takes his or her Nash equilibrium strategies, the false alarm rate will be 0. (b) The detection rate can be predicted using the Nash equilibia, as illustrated in Figures 6 and 7. Note that here the detection rate actually is $1 - attacker\_success\_rate$. However, (a) since there is incomplete information for the attacker to compute his or her exact Nash equilibrium strategies, the detection rate can only be approximately predicted; (b) since there could not be enough computation or communication resources for the good guy to update his or her Nash equilibrium strategies in real time (Note that each game play could be based on different values of $CL(c_i)$ and $P(c_i)$ which are dynamically changed), the false alarm rate is usually not zero. Note that our attack prediction models

do not conflict with real world false alarm rate and detection rate measures since our predictions are not closely followed in real world.

5 Optimization: Using Signaling Games to Predict Attacks

Till now, we assume each play of an APG is simultaneous and during each play no player knows the actions of the other players, however, this assumption could be too weak in some attack prediction scenarios. For example, in credit card authorization it is usually true that the ADS can observe the action of the attacker before taking an action. This ability actually can give the ADS some advantage to choose better actions (since the ADS now has more knowledge about the attacker), however, this advantage is not taken by our previous prediction models.

In this section, we present a signaling-game model to exploit this ability to do better defense. Signaling games have been used to analyze a variety of economic problems such as industrial organization [MR82], dividend policy [Bha79], management share ownership [LP77], job-market signaling [Spe73], corporate investment and capital structure [MM84], and monetary policy [Vic86]. In this paper, we model the attack-defense process as a sequence of (repeated) signaling games between a Customer and the ADS (on a credit card c_i), and each *signaling game* is a dynamic two-stage game with the following timing: (1) Nature draws a type t_i for the Customer from a set of feasible types $T_{subject} = \{t_1^g, t_2^b, ..., t_I^b\}$ according to a probability distribution $p(t_i)$, where $p(t_i) > 0$ for every i and $p(t_1^g) + ... + p(t_I^b) = 1$. (2) The Customer observes t_i and then chooses a credit card transaction T_j from a set of feasible transactions $TR = \{T_1, ..., T_J\}$. (3) The ADS observes T_j (but not t_i) and then chooses an threshold from a set of feasible thresholds $TH = \{th^1, ..., th^K\}$. (4) The payoffs are given by $u_{customer}(t_i, T_j, th^k)$ and $u_{ads}(t_i, T_j, th^k)$.

This signaling game satisfies the following requirements:

[Signaling Requirement 1] After receiving any transaction T_j from TR, the ADS must have a belief about which types could have sent T_j. Denote this belief by the probability distribution $p(t_i|T_j)$, where $p(t_i|T_j) > 0$ for each t_i in $T_{subject}$, and $\sum_{t_i \in T_{subject}} p(t_i|T_j) = 1$.

[Signaling Requirement 2] For each T_j in TR, the ADS's action $th^*(T_j)$ must maximize the ADS's expected utility, given the belief $p(t_i|T_j)$ about which types could have sent T_j. That is, $th^*(T_j)$ solves $\max_{th^k \in TH} \sum_{t_i \in T_{subject}} (p(t_i|T_j) u_{ads}(t_i, T_j, th^k))$.

[Signaling Requirement 3] For each t_i in $T_{subject}$, the Customer's transaction $T^*(t_i)$ must maximize the Customer's utility, given the ADS's strategy $th^*(T_j)$. That is, $T^*(t_i)$ solves $\max_{T_j \in TR} u_{customer}(t_i, T_j, th^*(T_j))$.

[Signaling Requirement 4] For each T_j in TR, if there exists t_i in T such that $T^*(t_i) = T_j$, then the ADS's belief about which types could have sent T_j must follow from Bayes' rule and the Customer's equilibrium strategy: $p(t_i|T_j) = p(t_i) / \sum_{t_i \in Type_j} p(t_i)$. Here $Type_j$ denotes the set of types that send the transaction T_j. That is, t_i is a member of $Type_j$ if $T^*(t_i) = T_j$.

The predicted attack is the set of $T^*(t_i^b)$ within the *perfect Bayesian equilibrium* of this game. A pure-strategy perfect Bayesian equilibrium in a signaling game is a pair of *strategies* $T^*(t_i)$ and $th^*(T_j)$ and a belief $p(t_i|T_j)$ satisfying Signaling Requirements (1) to (4).

In order to understand signaling attack prediction games in a more intuitive way, we exploit the *extensive-from representation* of games. This representation is a powerful analysis tool that can be used to formalize every attack prediction game proposed in this paper. Using this tool, we can clearly identify the inherent relationships among the variety types of attack prediction games presented.

The extensive-from representation of games can be visualized by a game tree. Each *game tree* begins with a *decision node* for player 1, where player 1 chooses an action. After player 1 chooses an action a_i, a decision node for player 2 is reached where player 2 can choose an action. These two decision nodes are connected by an arc marked by action a_i. To illustrate, let's consider a very simple attack prediction game. Assume $T_{subject} = \{good, bad\}$, $TR = \{T_1, T_2\}$ for both types, and

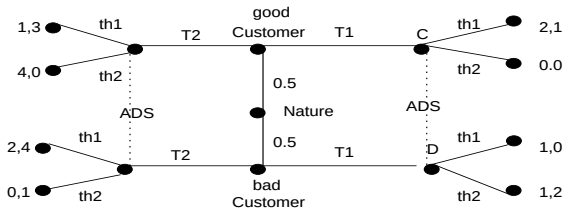

Figure 10: A Simple Signaling Attack Prediction Game

$TH = \{th^1, th^2\}$. Assume that $p(good) = 0.5$, then $p(bad) = 0.5$. Then the game tree of this signaling game can be shown in Figure 10. Note that the corresponding payoffs are also marked on the game tree. In each payoff pair (x, y), x indicates the payoff for the Customer and y indicates the payoff for the ADS.

Note that here decision nodes C and D are connected by a dashed line since when a *play* of the game reaches a node in the set $\{C, D\}$, the ADS has the *move* (i.e., can choose an action) but does not know which node in the set has (or has not) been reached (since the ADS does not know the type of the Customer). We call such a set an *information set* for the ADS. An information set with a single element is called a *singleton* information set. Note that the extensive-form representation of a game not only specifies the players, the strategies, and the payoffs, but also specifies (1) when each player has the move, and (2) what each player knows at each of his or her opportunities to move. Note also that the extensive-form representation of a game can easily specify simultaneous plays using the concept of information sets.

Compared with the static Bayesian attack prediction game proposed in Section 3, the ADS's belief is more accurate. In particular, in static Bayesian prediction games the ADS believes that every customer type could submit every kind of transactions, that is, the ADS believes that $P(t_i|T_j)$ is always equal to $P(t_i)$. However, in dynamic signaling games, after the ADS observes the action of the Customer and gains more knowledge, the ADS's belief can be made more accurate. For example, let's reconsider the game shown in Figure 10, it is not very difficult to figure out that in the Nash equilibrium of this game the good guy will choose T_1 and the bad guy will choose T_2. This indicates that if the ADS observes that the action of the Customer is T_1, then the ADS's belief should be $P(good|T_1) = 1.0$ instead of $P(good)$ which is 0.5 because according to the Nash equilibrium the attacker will never choose T_1. As a result, more accurate beliefs will certainly improve the prediction accuracy.

6 Enforcement Issues

A concern about game-theoretic attack prediction models is how to put these models into practice use. In this session, we address several enforcement issues.

Prediction accuracy issues. The prediction accuracy of our approach is dependent on several assumptions. (1) The rationality notion of an expected-utility maximizer should be accepted by both the attacker and the system. That is, both the attacker and the system will take rational actions. (2) When there are multiple Nash equilibria for a single game (play), the attacker and the system should be able to select the same Nash equilibrium. (3) Although both the attacker and the system usually have incomplete knowledge about the other player, their beliefs should be consistent with the real world situation. It is not difficult to see that these assumptions can be satisfied in a variety of real world attack prediction applications. Moreover, the discussions in Section 3 indicate that the prediction accuracy is also dependent on whether or not the predictions produced can be known to

(or accurately estimated by) the attacker and the system.

Computational issues. Game-theoretic attack prediction models usually involve computation intensive combinational problems. When the game is simple, e.g., the game that we have simulated, computing the Nash equilibrium strategies takes reasonable amount of resources. However, real world games are usually complicated. For example, to make a credit card fraud prediction game practical, we may need to handle more complicated ADS thresholds, profiles, and attacking actions, which can cover much more aspects (or attributes) of credit card transactions than transaction amounts. Practical games usually involve very large multi-dimension action spaces, so solving practical games can be extremely resource consuming, and practical game-theoretic reasoning usually needs parallel computing, relaxation, and approximations [PU00, EJK+00]. Fortunately, this problem is not the unique problem for attack prediction games but a common problem for almost every game theory application. The fact that game-theoretic models have been successfully used in e-commerce (e.g., online auctions) (through a variety of economical computation methods such as iterative computation, distributed computation, approximations, and bounded rationality) shows that the computation problem could be solved for practical game theoretic attack prediction. Practical computational game theoretic attack prediction is out of the scope of this paper.

7 Conclusion and Future Work

The ability to predict attacks can dramatically enhance people's capacity to defend cyber attacks. This paper - the first of a series - presents a novel game-theoretic attack prediction framework where the Nash equilibria of attack prediction games can generate valuable attack predictions, give a good estimation of the maximum security loss, and tell how the defense should be built. Our preliminary results on credit card fraud prediction are very encouraging. As part of the future work, we are investigating the enforcement issues in depth, and extending our APG model to incorporate the signaling game optimization and to predict other types of cyber attacks such as DDoS attacks.

References

[BAMF01] H. Browne, W. A. Arbaugh, J. McHugh, and W. L. Fithen. A trend analysis of exploitations. In *Proc. 2001 IEEE Symposium on Security and Privacy*, pages 214–229, May 2001.

[Bha79] S. Bhattacharya. Inperfect information, divident policy, and the 'bird in the hand' fallacy. *Bell Journal of Economics*, 10:259–270, 1979.

[EJK+00] J. E. Eggleston, S. Jamin, T. P. Kelly, J. K. MacKie-Mason, W. E. Walsh, and M. P. Wellman. Survivability through market-based adaptivity: The marx project. In *Proc. 2000 DARPA Information Survivability Conference (DISCEX)*, June 2000.

[FPS01] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[Gib92] R. Gibbons. *Game Theory for Applied Economics.* Priceton University Press, 1992.

[GL91] T.D. Garvey and T.F. Lunt. Model-based intrusion detection. In *Proceedings of the 14th National Computer Security Conference*, Baltimore, MD, October 1991.

[Har73] J. Harsanyi. Games with randomly disturbed payoffs: A new rational for mixed strategy equilibrium points. *International Journal of Game Theory*, 2, 1973.

[HRW00] J. Hu, D. Reeves, and H. Wong. Personalized bidding agents for online auctions. In *Proc. 5th International Conference on Practical Application of Intelligent Agents and Multi-Agents*, May 2000.

[HS88] J. C. Harsanyi and S. Selten. *A General Theory of Equilibrium Selection in Games.* MIT Press, 1988.

[IKP95] K. Ilgun, R.A. Kemmerer, and P.A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering*, 21(3):181–199, 1995.

[Ilg93] K. Ilgun. Ustat: A real-time intrusion detection system for unix. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 1993.

[JV91] H. S. Javitz and A. Valdes. The sri ides statistical anomaly detector. In *Proceedings IEEE Computer Society Symposium on Security and Privacy*, Oakland, CA, May 1991.

[JV94] H. S. Javitz and A. Valdes. The nides statistical component description and justification. Technical Report A010, SRI International, March 1994.

[LP77] H. Leland and D. Pyle. Informational asymmetrics, financial stucture, and financial intermediation. *Journal of Finance*, 32:371–387, 1977.

[Lun93] T.F. Lunt. A Survey of Intrusion Detection Techniques. *Computers & Security*, 12(4):405–418, June 1993.

[LX01] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proc. 2001 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2001.

[MCWG95] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1 edition, 1995.

[MHL94] B. Mukherjee, L. T. Heberlein, and K.N. Levitt. Network intrusion detection. *IEEE Network*, pages 26–41, June 1994.

[MM84] S. Myers and N. Majluf. Corporate financing and investment decisions when firms have information that investors do not have. *Journal of Financial Economics*, 13:187–221, 1984.

[MR82] P. Milgrom and J. Roberts. Limit pricing and entry under incomplete information. *Econometrica*, 40:443–459, 1982.

[Nas50] J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36, 1950.

[Pap94] L. Papayanopoulos. Preventing minority disenfranchisement through dynamic bayesian reapportionment of legislative voting power. In T. Basar and A. Haurie, editors, *Advances in Dynamic Games and Applications*, pages 386–394. Birkhauser Boston, 1994.

[PH01] D. C. Parkes and B. A. Huberman. Multiagent cooperative search for portfolio selection. *Games and Economic Behavior*, 35:124–165, 2001.

[PSC98] K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional qos vectors, and burstiness. In *Proc. International Conference on Information and Computation Economics*, 1998.

[PU00] D. C. Parkes and L. H. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artifical Intelligence*, pages 74–81, 2000.

[San00] T. W. Sandholm. Issues in computational vickrey auctions. *International Journal of Electronic Commerce*, 4:107–129, 2000.

[SBB01] S. Sekar, M. Bendre, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *Proc. 2001 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2001.

[SFL97] S. Stolfo, D. Fan, and W. Lee. Credit card fraud detection using meta-learning: Issues and initial results. In *Proc. AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, 1997.

[SL96] T. W. Sandholm and V. R. Lesser. Advantages of a leveled commitment contracting protocol. In *Proc. 13th National Conference on Artificial Intelligence*, pages 126–133, 1996.

[SM97] D. Samfat and R. Molva. Idamn: An intrusion detection architecture for mibile networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1373–1380, 1997.

[Spe73] A. M. Spence. Job market signaling. *Quarterly Journal of Economics*, 87:355–374, 1973.

[Vic86] J. Vickers. Signaling in a model of monetary policy with incomplete information. *Oxford Economic Papers*, 38:443–455, 1986.

[Vin94] T. L. Vincent. An evolutionary game theory for differential equation models with reference to ecosystem management. In T. Basar and A. Haurie, editors, *Advances in Dynamic Games and Applications*, pages 356–374. Birkhauser Boston, 1994.

[WW00] P. R. Wurman and M. P. Wellman. Akba: A progressive, anonymous-price combinatorial auction. In *Proc. 2nd ACM Conference on Electronic Commerce*, pages 21–29, 2000.

A A Simple Review of Game Theory

Definition 3 [Normal-Form Representation of Game] The *normal-form representation* of an n-player game specifies the players' *strategy* spaces S_1, ..., S_n and their *payoff* functions u_1, ..., u_n. We denote this game by $G = \{S_1, ..., S_n; u_1, ..., u_n\}$.

Definition 4 [Nash Equilibrium] In the n-player normal-form game $G = \{S_1, ..., S_n; u_1, ..., u_n\}$, the strategies $(s_1^*, ..., s_n^*)$ are a *Nash equilibrium* if, for each player i, s_i^* is (at least tied for) player i's best response to the strategies specified for the n-1 other players, $(s_1^*, ..., s_{i-1}^*, s_{i+1}^*, ..., s_n^*)$. That is, s_i^* solves $\max_{s_i \in S_i} u_i(s_1^*, ..., s_{i-1}^*, s_i, s_{i+1}^*, ..., s_n^*)$.

A *pure strategy* for player i is an element of set S_i. Suppose $S_i = \{s_{i1}, ..., s_{ik}\}$, then a *mixed strategy* for player i is a probability distribution $p_i = (p_{i1}, ..., p_{ik})$, where $o \leq p_{ik} \leq 1$ for $k = 1, ..., K$ and $p_{i1} + ... + p_{ik} = 1$.

Although many games have one or more pure strategy Nash equilibria, a game could have no pure strategy Nash equilibrium. Nevertheless, it is proved that when mixed strategies are allowed, every game has at least one Nash equilibrium.

Theorem 1 (Nash 1950): In the n-player normal-form game $G = \{S_1, ..., S_n; u_1, ..., u_n\}$, if n is finite and S_i is finite for every i then there exists at least one Nash equilibrium, possibly involving mixed strategies.

Definition 5 [Static Bayesian Game] The normal-form representation of an n-player static Bayesian game specified the players' *action* spaces $A_1, ..., A_n$, their type spaces $T_1, ..., T_n$, their beliefs $p_1, ..., p_n$, and their payoff functions $u_1, ..., u_n$. Player i's *type*, t_i, is privately known by player i, determines player i's payoff function, $u_i(a_1, ..., a_n; t_i)$, and is a member of the set of possible types, T_i. Player i's *belief* $p_i(t_{-i}|t_i)$ describes i's uncertainty about the $n-1$ other players' possible types, t_{-i}, given i's own type, t_i. We denote this game by $G = \{A_1, ..., A_n; T_1, ..., T_n; p_1, ..., p_n; u_1, ..., u_n\}$.

Definition 6 [Bayesian Nash Equilibrium] In the static Bayesian game
$G = \{A_1, ..., A_n; T_1, ..., T_n; p_1, ..., p_n; u_1, ..., u_n\}$, the strategies $(s_1^*, ..., s_n^*)$ are a (pure-strategy) *Bayesian Nash equilibrium* if for each player i and for each i's type t_i in T_i, $s_i^*(t_i)$ solves
$\max_{a_i \in A_i} \sum_{t_{-i} \in T_{-i}} u_i(s_1^*(t_1), ..., s_{i-1}^*(t_{i-1}), a_i, s_{i+1}^*(t_{i+1}), ..., s_n^*(t_n); t) p_i(t_{-i}|t_i)$.

Definition 7 [Finitely Repeated Game] Given a stage game G, let $G(T)$ denote the *finitely repeated game* in which G is played T times, with the outcomes of all preceding plays observed before the next play begins. The payoffs for $G(T)$ are simply the sum of the payoffs from the T stage games.

Proposition 1 If the stage game G has a unique Nash equilibrium then, for any finite T, the repeated game $G(T)$ has a unique subgame-perfect outcome: the Nash equilibrium of G is played in every stage.

Definition 8 [Multi-Stage Strategy] In the finitely repeated game $G(T)$, a player's *strategy* specifies the action the player will take in each stage, for each possible history of play through the previous stage.

Definition 9 [Subgame] In the finitely repeated game $G(T)$, a *subgame* beginning at stage $t+1$ is the repeated game in which G is played $T-t$ times, denoted $G(T-t)$. There are many subgames that begin at stage $t+1$, one for each of the possible histories of play through stage t.

Definition 10 [Subgame-Perfect Nash Equilibrium] A Nash equilibrium is *subgame-perfect* if the player's strategies constitute a Nash equilibrium in every subgame.