Optimizing Security Mechanism for Electronic Commerce (Preliminary Report)

XiaoFeng Wang Pradeep K. Khosla
Department of Electrical and Computer Engineering
Carnegie Mellon University

Ramayya Krishnan
Heinz School
Carnegie Mellon University

Abstract

The benefits of a security mechanism should be weighed against its cost to maximize system utility. However, traditional hard security techniques (cryptography) consider only the worst-case situation thus complicating security mechanism for electronic commercial transaction. On the other hand, though some soft security alternatives (game theoretic incentive engineering based on reputation mechanism) are attempted to realize less expensive self-enforcing security by engineering players' incentives, they tend to be either computationally intractable or sub-optimal. In our research, we consider the mechanism combining hard and soft security together to optimize the security strategies for the players in the Internet trading scenario. An example presented in this preliminary work is an approach to protect mobile trade agent from malicious merchant hosts. By using game theoretic model and non-repudiation tracing method, the mechanism provides an equilibrium strategy for consumers to protect their agents. Further investigations are made to investigate the possibility of pushing this Nash point to approach an optimal point by driving merchants' incentives. This approach is trying to keep the optimization and stability of the security strategy during the evolution of players' intentions.

1. Introduction

The worldwide expansion of network access is driving an increase in electronic trading activities. Both merchants and customers can benefit from the vast amount of commercial information online and convenient communication channel. However, such open trading environment also brings in great security challenges. Under the chaotic and uncertain Internet, a trader would find difficult to know his perspective partner's identity, to say nothing of protecting himself from being deceived.

Traditional approach to this problem is to use cryptography techniques. Security protocols help establish that the party you are dealing with is authenticated and authorized to take various actions. They also guarantee the integrity and confidentiality of the data in order to detect possible violations of the trading agreements and non-repudiation of activities to trace the malicious breeders. Some security commercial transactions, such as SET **[1]**, have already been implemented in the electronic commerce. Other lower layer protocols such as PGP, IPSec and SSL are also widely used in commercial activities.

A major problem of the cryptography approach (we call it *hard* security) is its cost. Hard security usually considers the worst-case situation. It assumes that the potential enemies have perfect information and organization and are irrational, sparing no efforts to commit crimes. This assumption makes security protocols complicated (e.g., SET) and in some cases, even intractable (e.g., mobile code security).

As a complement to hard security, *soft* security takes players' rationalities into consideration. It uses game theoretic engineering to reduce players' evil incentives to realize a low cost, self-enforcing security. Underneath the soft security approach is the idea of *mechanism design*. Previous works **[2][3]** on this domain are to establish communication protocols basing on players' rationalities. Important approaches directly relating to secure electronic commerce include safe exchange protocols **[4][5][6]**. In this problem domain, seller and buyer want to realize a contract-free delivery. Each party has the danger to be cheated by and the intention to cheat her counterpart. A soft security solution is to divide the good and payment into small chunks and exchange them sequentially so that both parties will gain more by completing the transaction than stopping in the middle way and vanishing. Another related approach is using ecologic systems **[7][8][9]** to put social controls on the electronic traders (*Intelligent Trade Agents*). Players behaving undesirably will finally be eliminated from the market. These manifest that soft security has potentials to establish a security mechanism with lower cost (self-enforcing) and more robust (evolvable) than its hard alternative.

However, pure soft security also has its limitation. Relying on the penalty and reward to regulate players' intentions would become computational intensive and even intractable in certain cases. The researchers in safe exchange protocols find it difficult to guarantee the security of transaction without external costs in some cases **[5]**. On the other hand, without the help of hard security, a game theoretic mechanism cannot wash out all the irrational players who can produce fake identities. The third problem for soft security is how to penalize the malicious breeders. Under the constraint of different legislation systems, it is difficult to punish an evil trader in the foreign country. A possible solution to this difficulty is to use trust and reputation management **[7][8][9]**. A bad guy will suffer a loss of reputation once his misbehaviors are discovered. This will result in a future loss of his profits. In our research, we take reputation as the major channel to issue awards and penalty.

In our research, we consider combining soft and hard security together to optimize security mechanism for electronic commerce. Our approach consists of two stages. In the first stage, the security mechanism takes all the players as rational entities. Hard security measure will be traded off between its benefits (save of losses) and costs based on the analysis of players' incentives and divulgence of information to optimize their expected profits. In the second stage, we study the possibility of reducing players' malicious intentions in an evolutional way. As a first step to this research, we attack this general problem through a specific case which tries to find an optimal solution to protect mobile trade agent from malicious merchant hosts. We will extend our results from this specific study to more general situations in the future works.

The rest of the paper is organized as follows: In Section 2, we introduce the problems and difficulties to protect mobile trade agents, and then present our model in Section 3; Further discussion on the possible evolution of players' incentives will be given in Section 4; The last section will conclude the works and point out future researches.

2. Case Study: Protecting Mobile Trade Agents

Since security problem inside electronic commerce is too general to tackle, we start our research with a specific case: mobile code security. The results from this study would be implemented in other areas of e-commerce after revisions.

Mobile agent is small software capable of working on people's behalf. In the scenario of mobile agent based Internet trading, agents can jump through traders' computers to collect merchant or product information, negotiate with merchants and even conduct offline payment for their owners. This reduces latency, balances network loads, encapsulates protocols and makes trading system very flexible and robust **[10]**.

The security difficulties in the mobile agent systems are from two perspectives: how to protect servers from virus agents and how to protect agents from evil-intended merchant hosts **[10][11]**. The former is an extending fear of viruses. In an open system, it is dangerous if you cannot verify the functionalities of a foreign program running on your computer. Fortunately, mature techniques on the virus protection help people to overcome the difficulty. A successful approach is to use sandbox and proxies to constrain the activities of untrusted agents **[12]**. The latter problem, however, produces a great challenge to the researchers in system security. This is because theoretically, any finite automata can be traced, controlled and modified by its host computer. Although many approaches are proposed to attack this difficult problem **[12][13][14]**, so far, there is no applicable solution.

In our research, we consider the problem of protecting mobile trade agent in a simplified trading setting. It can be described as follows:

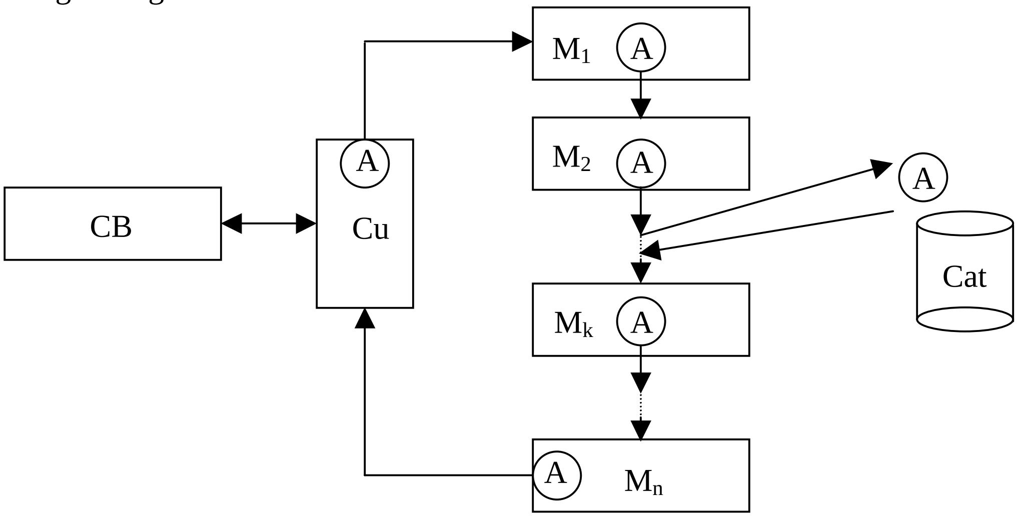


Figure 1. Mobile agent trading

In Figure 1, *A* stands for a mobile trade agent; M_k for a the *k*th merchant (it can be taken as a merchant agent who can take controls of the mobile trade agent when it gets into the merchant server); *Cu* for customer (in a similar way, it can also be taken as a static customer agent who is working for a human customer); *CB* is a credit bureau; *Cat* for online catalogues.

In the trading process described in the above figure, customer agent sends out a mobile trade agent who itinerates merchant hosts to gather information about products according to customers' interests. The mobile agent can plan or meta-plan his travel route through analyzing the merchant/product information from the online catalogues. During this outsourcing process, a malicious merchant agent could take control of the mobile agent after he got into the merchant's host. Illegal actions can be taken to delete some quotations the mobile agent gathered from other merchants, or revise his itinerary plan to avoid some strong competitors. The objective of the merchant agent is to distinguish itself as the best vendor according to the customer's tastes. These behaviors are totally blind to the customer. Thus, he has no idea about whether the returning purchaser is trustworthy.

A hard security measure of protecting the mobile agent is to enable the merchants to keep all the non-repudiation evidence about a passing-by agent for a period of time. Such evidence includes a mobile agent's identity/certificates, digital signatures on the digest of the agent's codes and data, information agent gathered from current merchant, the next merchant agents moved to, the acknowledge signature on the agent from the next merchant and time stamps. All the information can be kept in a merchant database called Login Data Base (LDB) **[15][16]** and updated periodically. With such evidence, a suspecting customer agent can launch an investigation to validate the final results.

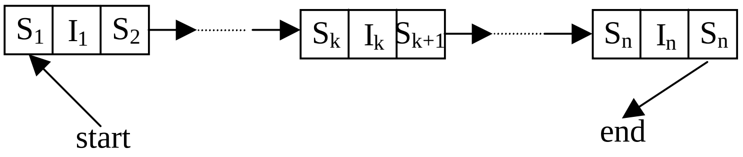


Figure 2. Tracing the malicious merchant

The above figure shows the tracing process. Starting from the first server mobile agent moved to, a merchant agent can trace and validate the whole trading process by analyzing the mobile agent's behaviors and non-repudiation evidence from merchants. This method can help to dig out potential illegal actions **[16]**. However, it is very expensive, and thus cannot be used casually.

To indicate to customer how trustworthy trading results are, credit bureau is an agency who assigns a reputation value to every merchant. If the merchant selected by the mobile agent is highly reputable. The customer agent has few reasons to suspect the results. On the other hand, a notorious seller will be distrusted by customer and suffer an investigation. This will help to find a balance between the benefits of the hard security measures (tracing) and its costs.

In the next section, we will analyze the security strategy of customer in a formal way. A game theoretic model will be used to optimize the security mechanism.

3. Formal analysis on the security mechanism

The mobile agent based trading system in Section 2 can be formally described as a 8-tuple $<M, Cu, V, L, R, F, G, C>$, where

- M is a set of merchant agents.
- Cu is a set of customer agents.
- $V: M \cup Cu \rightarrow \Re$ is a utility function which maps a merchant to his utility to sell out a product and a customer to his utility to buy a satisfied product. For $m \in M$, $V(m)$ is denoted as V_m; for $c \in C$, $V(c)$ is denoted as V_c.
- $L: M \cup Cu \rightarrow \Re$ is a loss function which maps a merchant to his loss coming from loss of reputation once his malicious actions are detected; and a customer to his loss if he was cheated to buy an unsatisfied product. For $m \in M$, $L(m)$ is denoted as L_m; for $c \in Cu$, $L(c)$ is denoted as L_c.
- $R: M \rightarrow [0,1]$ is a reputation function which maps a merchant to his reputation value given by the credit bureau. Such reputation value is represented by a probability value. Intuitively, it denotes the probability that the merchant will not attack a mobile trade agent to cheat a customer. For $m \in M$, $R(m)$ is denoted as R_m.
- F is a probability distribution (c.d.f) of V_m. $f(.)$ is its corresponding p.d.f.
- G is a probability distribution c.d.f of L_c. $g(.)$ is its corresponding p.d.f.
- C is customers' costs to trace the trading procedure.

In this trading system, merchants know their own utility V_m, losses L_m, reputation R_m, customers' loss distribution G and cost in tracing C; customers know merchants' reputation, his own utility V_c and security cost C; credit bureau knows a merchant's loss (how much penalty he can put on the merchant through adjustment of his reputation R_m), security cost C and distribution F and G.

We model the trading process as a repeated 2-person sequential game. During the trading process, merchants (exclude the final winner) first make decision on whether to defect (tamper the mobile agent) or not, and then a customer make decision on whether to trust the trading result (buys the product from the winning merchant) or distrust it (starts an investigation)[1]. This can be described as the following game tree:

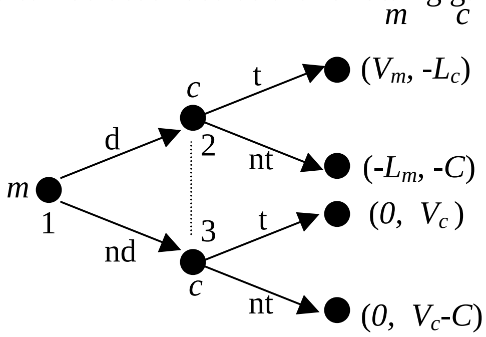


Figure 3. Game tree of the trading process

In Figure 3, merchants (except customer's ideal seller) make decisions on defecting (*d*) or not defecting (*nd*) at node 1. After the mobile agent returned, customer has to decide

[1] To simplify the research setting, we assume customers only can take these two actions. Actually, a real customer may simply discard the trading result and start a new round when she suspects the results.

whether to trust (*t*) the trading results or not (*nt*). The problem is that customer does not know which nodes he is at (2 or 3). Node 2 and node 3 forms the customer's information set. At node 2, if the customer trusts the result, the malicious merchant will get profit V_m and the customer will suffer a loss $L_{c;}$ if customer distrust results and start a tracing, he can avoid the loss but has to pay a cost *C* while the merchant will be dig out and suffer a great loss L_m through the defaming of reputation. At node 3, the merchant can get nothing no matter what actions the customer takes because he is not the ideal seller for the customer; for the customer, if he trusts results, he can get a utility V_c and if not, he has to pay a tracing fee which cut down the final utility to V_c-*C*.

The customer agent knows the winning merchant's reputation value R_m which indicates his probability of being at node 3. Thus, the customer agent can make decision by calculating his expected profits if he trust the result

$$\hat{V}_c^{t} = R_m V_c - (1 - R_m) L_c \tag{1}$$

and then comparing it with the expected profit if he distrust the result

$$\hat{V}_c^{nt} = R_m (V_c - C) - (1 - R_m) C \tag{2}$$

If

$$\hat{V}_c^{t} \geq \hat{V}_c^{nt} \tag{3}$$

he will trust the results; or else, he will start investigation.

After simplified (3), we can get the customer's strategy as follows:

Definition 1 *Customer's strategy C_s: If $L_c \leq \frac{C}{1 - R_m}$, trust the merchant and buy the product, or else distrust him and start a tracing process.*

From a merchant's point of view, he knows his reputation R_m and the tracing cost *C*, and thus he can calculate $\frac{C}{1 - R_m}$. However, he has no exact information about the customer's losses L_c. Actually, he only knows the probability distribution of L_c which is represented by a *c.d.f F(.)*. Thus, he can estimate the probability of being undetected after committed a crime as:

$$P_s = P\{L_c \leq \frac{C}{1 - R_m}\} = F(\frac{C}{1 - R_m}) \tag{4}$$

The merchant can figure out his expected utility if defects:

$$\hat{V}_m^{d} = P_s V_m - (1 - P_s) L_m \tag{5}$$

Suppose the merchant cannot get the transaction without tampering with the trade agents (He is not the most desirable seller according to buyer's preference). Therefore, he will decide not to defect only when $\hat{V}_m^d \leq 0$. After simplified this inequality, we have:

Definition 2 *Merchant's strategy M_s: If $V_m \leq \frac{(1-P_s)L_m}{P_s}$, behave honestly, or else tamper with the mobile agent.*

The credit bureau does not know an individual merchant's utility V_m but knows its distributions *G(.)*. So, if the merchant is randomly drawn from this distribution, his probability to defect should be:

$$P\{V_m \leq \frac{(1-P_s)L_m}{P_s}\} = G\{\frac{(1-P_s)L_m}{P_s}\} \tag{6}$$

This leads to following theorem:
Theorem 1 *If merchant m's utility is randomly drawn from G(.) and customer c's loss L_c is from F(.); and if $R_m = G(\frac{(1-P_s)L_m}{P_s})$, the strategies C_s and M_s form a Nash equilibrium for the repeated game in the Figure 3 .*

Nash equilibrium means that each player is motivated to abide to its specified strategy given that the other abides to its specified strategy. For the repeated trading game described in Figure 3, with the divulgence of information about *F(.)*, *G(.)*, *C*, L_m and rationality of merchants, C_s becomes a customer's optimal security strategy because it maximizes his expected utility.

To establish such equilibrium, the credit bureau only needs to assign each merchant a credit value by solving the equation:

$$R_m = G(\frac{(1-F(\frac{C}{1-R_m}))L_m}{F(\frac{C}{1-R_m})}) \tag{7}$$

For example, suppose both *C* and L_m are between 0 and 1, and *F(.)* and *G(.)* are uniform distribution between 0 and 1, we can solve the equation (7) and find R_m as:

$$R_m = \frac{(1-C)L_m}{C+L_m} \tag{8}$$

Figure 4 shows the change of the R_m with L_m, which denoted as $T(L_m)$:

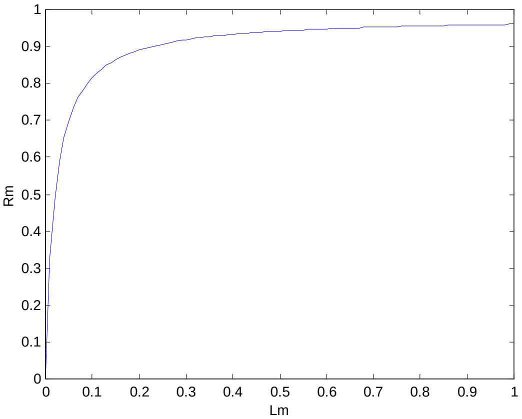


Figure 4. Example of Reputation *$T(L_m)$* with *C*=0.02

The figure shows that with fixed tracing cost *C*, merchant *m*'s reputation R_m is a concave increasing function of his losses L_m. This is in accordance with intuitions: the more losses a merchant will suffer from the loss of reputation, the less likely that he will commit a crime. Therefore, with the known *C*, customers could trust a merchant according to his potential losses due to the defaming.

4. The possibility of reputation evolution

Section 3 analyzes the optimal security strategy customer agents can take to protect their mobile agents given merchant agents' rational behaviors. Actually, such strategy is optimal under the constraint of merchant agents' actions. That is, with the change of merchants' actions, customers' expected utilities might also change. In the ideal situation, when all the merchants are absolutely honest (with *R*=1), the customers' utilities are maximal because they need not spend money on hard security protection and are free of being cheated by merchants. Unfortunately, such situation is unstable because once customers stop their efforts on hard security (tracing), rational merchants' optimal strategies would become cheating. A more realistic setting is that with reputable merchants in majority, customers get better utilities than they do when notorious merchants in majority. This implies that a security mechanism should not only consider the optimal protection strategies customers can take given merchants' behaviors, but also the methods to make merchants' behaviors desirable.

One way to engineer merchants' incentives (in fact, it may be the only way under the Internet scenario) is by adjustment of their reputations. With a higher reputation value, a merchant would have more chances to be visited by mobile agents since customers may deliberately avoid those with low reputation values. This would increase his chances to get profits. Thus, it is reasonable to assume that merchants' profits are non-decreasing functions of their reputation values. We denote merchant *m*'s profit-reputation relation function as h_m(.).

Actually, h_m(.) is highly related to *m*'s losses L_m. L_m denotes *m*'s loss of profits in a period of time due to the loss of reputation, i.e., $L_m=h_m(r)-h_m(0)$. Suppose $h_m(0)=0$ (none will buy product from an ultimately notorious merchant). We can get $L_m=h_m(r)$. On the other hand, $r=T(L_m)$ (Figure 4). Thus, we have

$$h_m^{-1}(L_m) = T(L_m)$$

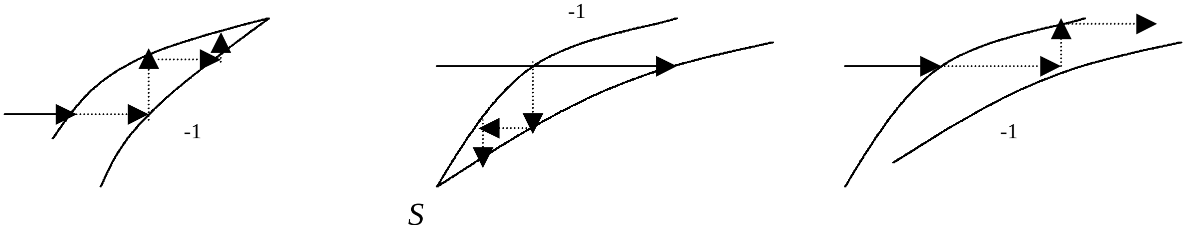
where, h^{-1} is the reverse function of *h*. This means (L_m,r) is an intersection point of function h_m^{-1} and *T* . We call it a *stable point*[2]. We also take the highest possible reputation value as such a *stable point*.

In real life, it would be difficult to estimate the function h_m (How much money you can get if I raise your reputation?). Therefore, if no one knows the function h_{m} (even merchant *m* himself), how can credit bureau to assign merchant *m* a reputation *r*? To solve the problem, we use a mechanism:

Definite 3. *Mechanism M1: 1.when new merchant m entering the market, assign him an initial credit value according to heuristic knowledge; 2. at the end of a period (e.g., 1 year), update m's reputation as: r=T(i), where i is m's earnings in this period*[3]*; 3. repeat 2 until r becomes stable or reaches the highest possible reputation value, i.e., reaching a stable point.*

Theorem 2 *Mechanism M1 can guarantee to find a stable point for merchant m.*

The following figures show an illustration to the above theorem.



(a) (b) (c)

Figure 5. Illustration of Theorem

Figure 5 shows three possible relations for h^{-1} and *T*. In (a), h^{-1} is lower than *T* and there is a stable point *S* at the top; in (b), h^{-1} is above *T* and the stable point is at the bottom; (c) shows the case that h^{-1} is below *T* and there is no upper intersection. In both (a) and (b), the repeated implements of *M1* guarantee the convergence to the stable point; in (c), the final results reach the maximal possible reputation value (it is also a stable point). Actually, the convergence process is similar to that in the cournot game.

[2] Actually, such stable points can be viewed as Nash equilibrium points in a game (similar to cournot game) between credit bureau and merchants.

[3] We suppose merchants' earnings are observable to the credit bureau.

Corollary. *In the trading system described in Section 2, if for any $m \in M$, there is a real number $0<o<$(the maximal losses merchant can afford) such that $T(x)>=h^{-1}(x)$ once $x>=o$, the optimal strategy to maximize customers' utilities is to assign merchant m the highest possible reputation value.*

According to (c) of Figure 5, the final results will converge to a maximal possible reputation value.

The strategy used in Corollary is to trust every merchant to the maximal possible level at first, if anyone is detected to default such trust, he will be severely punished. This minimizes the costs to enforce the security while still keeps the equilibrium in Section 2 under the pre-conditions of the corollary. In real life, such strategies are implemented in certain situations. In the bus system of Singapore and subway in German, it is a passenger's responsibility to buy the right ticket (fully trust the passenger at first). There are slim (but not zero) chances that a checker will check the passenger's ticket. Once the passenger is found to be cheating (did not buy or bought an under-paid ticket), she will be fined highly. Thus, a rational passenger is tending to buy a right ticket.

Unfortunately, in the Internet trading environment, it is difficult even to estimate an upper bound of a merchant's *h*. On the other hand, it is rare case that all the merchants' *h* can meet with the conditions of the Corollary. Thus, we need to consider a more general case.

Actually, though *M1* guarantees the convergence of the security strategy to stability, such evolution does not guarantee the optimal results. For example:

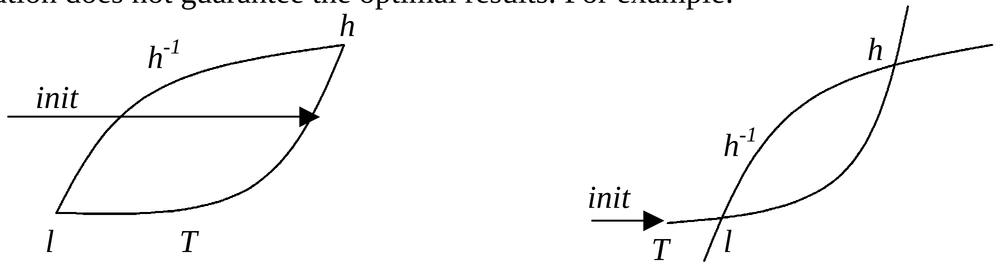


Figure 6. Example

In the above figure, the left one shows that if the initial value is introduced between two intersections and $h^{-1}>T$ between *h* (higher point) and *l* (lower point), the result will converge to *l* according to (b) of Figure 5. The right case will also converge to *l* according to (a). Therefore, merchants cannot attain their higher reputation values. This also reduces the customers' expected utilities because the security mechanism does not drive merchants' behaviors to the desirable.

To drive merchants out of lower stable points, a possible method is to let them bid. Merchant *m* can leave current stable point by estimating a reputation value and apply it to credit bureau. The problem is, if so, the optimal strategy *m* can use is to bid the highest possible reputation value. This can converge to his highest stable point by using *M1*. In such case, *m* has no intention to accelerate the convergence, which actually, increases customers' chances to be cheated thus reducing their expected utilities. To let merchant be more responsible for a fast convergence, we definite the following mechanism:

Definition 4. *Mechanism M2: 1. Each merchant estimates her higher stable point and bid the corresponding reputation value r to credit bureau; 2. credit bureau increases the merchant's reputation to his bid for a period (for example 1 year), and during such period, he will conduct irregular checks on the merchant's behaviors through tracing (hard security), whose costs will be put on the merchant; 3. by the end of the period, if the merchant's earnings (observable by the credit bureau) e and C(total tracing costs) make T(e+C)>=r, lets r=T(e+C); or else, with r=T(U), update r=T(2e+2C-U).*

The idea is, if the merchant under-bids, his reputation will be set according to his real earnings; if the merchant over-bids, he will be punished by a loss in reputation, whose value corresponds to the amounts of earnings that he promised but did not meet with (i.e., *U-(e+C).*). Thus, the merchant would have no intentions to over-bid, because he will be punished; he may not be willing to under-bid, because this may delay his chance to get a high profits thus resulting in the loss of his expected profits. His optimal strategy is to bid the amount that exactly enables him to jump to a higher stable point. An irregular checking of credit bureau makes merchants difficult to conduct cheating by taking advantage of their temporary reputation values.

The rest of the problem is, if merchant *m* himself does not know his profit-reputation function *h*(.), what kind of strategy he can take? What influence of such strategy on the general optimization of the system? A possible idea is to use learning technique. A merchant may make random tries at first, and then take further actions according to the rewards or punishments he received. A merchant may also be able to accelerate the learning process by observing and analyzing other mechants' experience. We will further investigate this problem theoretically and empirically.

5. Conclusion

In this paper, we studied the problem of optimizing security mechanism for electronic commerce systems. We are trying to combine both hard security (cryptography) and soft security (game theoretic incentive engineering) together to find an optimal trade-off between the benefits of security mechanism and its costs. Our approach is: 1. Instead of only considering a worst-case situation as in traditional hard security, we take into consideration the rationalities of players under the e-commerce scenario. This enables us to design optimal and stable security mechanisms basing on players' rational behaviors (Nash Equilibrium). 2. Incentive engineering mechanisms are used to drive players' behaviors to the desirable, and thus further push the Nash point towards the optimal point.

To avoid exploring the problem in a pure abstract way, we tackle the general problem ("optimizing security mechanism for electronic commerce") through a specific security problem (protection of mobile trade agents from malicious merchant hosts). The research results are expected to be extended to other problem domains. Actually, we keep an eye on the generality of the research during studying the specific problem. The formulation is possible to be converted to a general case. For example, the tracing cost *C* can be used to represent the general expense of implementing hard security. Actually, the research results are possible to be applied to other problems such as guaranteeing the stability of coalition among self-interested trade agents.

However, our current research is still in a preliminary stage. Further efforts are required to make the scheme complete. For example, we will design learning approaches to enable merchants to evolve to upper stable points quickly. Even further efforts should be made to practically apply the results to other problem domains. We study the generality through the specific, but the specific is only a start for the generality.

6. Reference

[1] Visa International, and Mastercard International. Secure Electronic Transaction (SET) Specification. Version 1.0, May 1997.

[2] J.S. Rosenschein and G. Zlotkin. Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. MIT Press, 1994.

[3]Dov Monderer and Moshe Tennenholtz. Distributed Games: From Mechanisms to Protocols. Sixteenth National Conference on Artificial Intelligence. AAAI-99.

[4] Sandholm, T. and Ferrandon, V. 2000. Safe Exchange Planner. International Conference on Multi-Agent Systems (ICMAS), Boston, MA, July 7-12.

[5] Sandholm, T. and Lesser, V. 1995. Equilibrium Analysis of the Possibilities of Unenforced Exchange in Multiagent Systems. 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, pp. 694-701.

[6] S.Matsubara, M.Yokoo. Defection-Free Exchange Mechanism for Information Goods. International Conference on Multi-Agent Systems (ICMAS), Boston, MA, July 7-12.

[7] Lars Rasmusson and Sverker Janson. Simulated social control for secure Internet commerce. In Proceedings of the Workshop on New Security Paradigms, 1996.

[8] Micheal Schillo and Petra Funk. Who can you trust: Dealing with deception. In Proceedings of the workshop Deception, Fraud and trust in Agent Societies at the Autonomous Agents Conference, Pages 95-106, 1999.

[9] Bin Yu and Munindar P. Singh. A Social Mechanism of Reputation Management in Electronic Communities. Cooperative Information Agents (CIA-99).

[10] Chess, D.M., Harrison, C.G., and Kershenbaum,A. Mobile agents: Are they a good idea? Research Report, IBM Research Division, T.J.Watson Research Center, Yorktown Heights, NY. Mar., 1995.

[11] J.Tardo and L. Valenta. Mobile agent Security and Telescript. In Proceedings of IEEE COMPCON’96, Feb. 1996.

[12] Vigna, G. Mobile Agents and Security. Springer-Verlag, Berlin,1998.

[13] X. Yi, X.F. Wang, K.Y. Lam. "A Secure Intelligent Trade Agent System", Trends in Distributed Systems '98: Electronic Commerce, Hamburg, Germany, LNCS, Springer-Verlag, June 3-5 1998.

[14]. X.F. Wang, X. Yi, K.Y. Lam and E. Okamoto. "Secure information gathering agent for Internet Trading", 11th Australian Joint Conference on Artificial Intelligence (AI'98), Brisbane, Australia, 13 July 1998, Springer-Verlag Lecture Notes in Artificial Intelligence, Vol. 1544, edited by Chengqi Zhang and Dickson Lukose, Springer-Verlag Publishers, pp. 183 -- 194, 1998.

[15] X. Yi, X.F. Wang, K.Y. Lam. "An Intelligent Agent Architecture for Securing Internet Trading", IFIP/SEC'98, 14th International Information Security Conference, 31 August 1998 - 4 September 1998.

[16] X.F. Wang, X. Yi, K.Y. Lam, C.Q. Zhang and E. Okamoto. "Secure Agent Mediated Auction-like Negotiation Protocol", CIA-99, Uppsala, Sweden, July 31 - August 2, 1999, LNAI, Vol. 1652, Springer-Verlag, 1999 (July), pp. 280 -- 291.