Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2006 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2006

Risk of Using Pirated Software and its Impact on Software Protection Strategies

Samuel Kwan
Samuel.Kwan@ust.hk

Jeevan Jaisingh
jeevan@ust.hk

Kar Yan Tam
kytam@ust.hk

Follow this and additional works at: http://aisel.aisnet.org/pacis2006

Risk of Using Pirated Software and its Impact on Software Protection

Strategies

Samuel Kwan
Department of Information
and Systems Management
Hong Kong University of
Science and Technology
Samuel.Kwan@ust.hk

Jeevan Jaisingh
Department of Information
and Systems Management
Hong Kong University of
Science and Technology
jeevan@ust.hk

Kar Yan Tam
Department of Information
and Systems Management
Hong Kong University of
Science and Technology
kytam@ust.hk

Abstract

The software protection strategy of software developer and the inherent risk to end user in using pirated software are two major factors that affect a user's decision on whether to purchase or pirate a software product. This paper analyzes the optimal protection strategy for software developer in horizontally and vertically differentiated markets. We find that the implementation cost of software protection constitutes the primary factor for software developers to determine their software protection strategies. However, in a vertically differentiated market, the lower quality product should always adopt a non-protection strategy, regardless of the protection implementation cost. In other cases, protection would only be optimal if the protection implementation cost to the software developer is relatively small. These findings are consistent with anecdotal evidence.

Keywords: Software piracy, software protection strategy, horizontal differentiation, vertical differentiation

1. Introduction

According to a global report by Business Software Alliance (2003), the software industry is said to have lost more than $13.08 billion in business during the year 2002 due to software piracy. It remains to be one of the most well known and persistent problems of the IT industry. It is estimated that an average of 39% of all software installed in 2002 were pirated versions (Business Software Alliance 2003). In fact, the situation is deteriorating with the rapid and pervasive application of IT in the modern society.

At the first glance, one may perceive software piracy as just another example of the illegal duplication of other intellectual properties like books, journals, or media contents. However, there are at least two factors that make software piracy a unique type of problem in itself. Firstly, software developers are able to implement protection codes or mechanisms that can effectively and significantly increase the difficulty to pirate. Such measures are usually not available to publishers and producers. The protection of software products can range from the use of hardware "dongles" to just requiring the user to enter a cryptic product key code. Although one may argue that these schemes are usually not unbreakable, they increase the cost for pirates to use illegal copies of software. Of course, the software developer would also need to pay a certain implementation cost for this type of software protection mechanisms. Secondly, there are inherent risks in using pirated software. These risks are unique to the software industry and make software piracy quite different from other

forms of illegal duplication of intellectual properties. For instance, due to the unlawful nature of piracy, the quality of pirated software can never be guaranteed. Very often the pirated software would have already been infected with malicious computer virus before it arrives at the hands of the end user.

Also, software is seldom free of defects. Some defects not only result in malfunctioning of the software but also leave "vulnerabilities" for malicious computer hackers to exploit. Very often these defects would only be discovered after the consumer has purchased and used the software for a certain period of time. It is a common practice for software developers to provide post-sale technical support to the legitimate users should there be any subsequent problems. While these types of technical support may exist in a variety of forms, ranging from security patches, features updates, to email or even telephone support, they are usually available only to the paying consumers but not to the pirates. Users of pirated software are at their own risk. It is reasonable to assume that some people would be deterred from pirating in view of this risk.

Cheng et al (1997) attempted to exhaustively enlist the reasons behind people's software purchasing or pirating decisions in an empirical study. In particular, they found "technical support in case of problems" as well as "worry about computer viruses" to be among the ten most important reasons for people to purchase, rather than pirate software.

In the Internet age, the risk of using vulnerable software cannot be overemphasized. The US-CERT Vulnerability Notes Database (US-CERT 2004) lists many popular software products that contain critical security vulnerabilities. For example, the various "buffer overrun" vulnerabilities that exist in the popular Oracle 9i product would require software updates available only to registered users. In principle, users of pirated copies of Oracle 9i would have to bear the risk of future attacks exploiting these vulnerabilities, especially if their systems are connected to the Internet. The risk in using pirated software became quite apparent in the recent "Blaster" attack. "Blaster" was an Internet virus that exploited a known vulnerability of the Windows XP software. A timely patch to this vulnerability was actually released by the vendor but was made available only to registered customers. As a result, machines running pirated copies of Windows XP were forced to disconnect from the Internet to prevent from getting exposed to the virus attack.

To summarize, both the software protection mechanism and the risk in using pirated software serve to reduce piracy. The former is mainly a preventive measure that is available to the software developer who needs to weigh the cost and benefit before deciding on what type of protection mechanism, if any, should be adopted. The latter can be considered a deterrent means that is beyond the control of any single party. The effect of risk may depend on a number of factors such as the chance of virus infection through the use of pirated software, the ultimate reliability of the software product[3], etc. In this paper, we attempt to study the optimal protection strategies for software firms in a duopoly software market, with the explicit consideration of the potential risk in using pirated software due to the lack of technical support. We develop an analytic model that assumes that people can overcome the software protection by bearing a cost of pirating. Users are heterogeneous in the cost of pirating. We will also model the risk in using pirated software in our framework.

We consider a duopoly market under two forms of product differentiation: horizontal and vertical (Tirole 1988). By horizontal differentiation, we mean that software firms compete with each other by designing features that are unique from its competitor. On the

---

[3] In this study, we assume that software developers would not intentionally reduce the reliability of their software products for the purpose of deterring piracy.

other hand, in a vertically differentiated market, firms compete in the quality of their products. Our main objective is to analyze the optimal protection strategies for software firms under each situation. The rest of the paper is organized as follows: To begin with, we will examine the relevant prior research in the area of software piracy first. We will then develop an analytical model conforming to our observations mentioned above. In particular, we will analyze the problem using two well known market competition paradigms, namely the horizontally differentiated market and the vertically differentiated one.

2.   Background Literature

Intuitively thinking, the existence of piracy should reduce demand and thus profit. It follows that the best strategy for software firms would be to increase the protection level as much as possible so that potential pirates would simply find it very costly to pirate. However, Conner & Rumelt (1991) argue that by taking into account the effect of network externalities, as commonly found in software products, a high level of protection may not be optimal for firms. They show that in a monopoly setting, raising software protection would be profit-maximizing only when there is only insignificant effect of network externalities. Otherwise, profit would decrease with an increased protection level because some would-be pirates are forced to do without the software, rather than buying it. However, the model used does not consider strategic interactions in a competitive market as well as the inherent risk in using pirated software.

The work by Shy & Thisse (1999) further develops the analysis of optimal software protection strategy by the use of a horizontally differentiated duopoly model. Their findings are similar to those by Conner & Rumelt in that when externalities effects are strong, non-protection would be optimal for competing software firms. Their model assumes dichotomization in two dimensions. Firstly, consumers are dichotomized into support-oriented or support-independent ones. Support-oriented users choose to buy rather than pirate if the price of software is less than the utility they derive from the support service. Secondly, rather than treating the level of protection level as a continuous variable, firm's protection strategies are dichotomized into either full protection or nil protection. This conforms to the reality in that software products usually come only with a simple protection mechanism that checks for a valid product key, or are just completely unprotected.

However, their model also assumes that end users would have no way to use pirated software if software protection is in place. Unfortunately this is often not true in real life. In fact, end users can usually obtain pirated software with protection mechanism already compromised. Also, most of the time software is only protected by a product key code that can be easily duplicated. Therefore, we believe that it should be more realistic to assume that people can still pirate even though a software product is protected, only that they would need to pay a cost of pirating. The cost for pirating can differ by individual but in general should be small compared with the price of software.

The dichotomization of support-oriented and support-independent consumers does not fully represent the inherent risk in using pirated software as discussed previously. In Shy & Thisse's model, technical support only brings a fixed amount of utility to support-oriented users. However, we believe the lack of technical support would introduce a risk that essentially discounts, in the sense of expected utility, the benefits derivable from using the pirated software.

The concept of risk in using pirated software was first used by Banerjee (2003) in his analysis of the software piracy problem from the perspective of social welfare and government policy. In his model, end user simply cannot be certain that a piece of software

obtained from illegal sources would indeed work as expected. However, his focus was mainly on governmental monitoring of counterfeiting business.

We believe that the concept of risk is not only applicable to cases when end user purchase software from pirates, but also to cases when end users make illegal copies of licensed software themselves. In fact, as pointed out earlier, the study by Cheng et al (1997) reveals that there are other risks in using pirated software, such as the lack of technical support in case of problems and computer viruses. In the following, market competition, dichotomization of software protection level, as well as the risk in using pirated software will be incorporated into our analytical model.

3. Horizontal Differentiation

We start with the horizontally differentiated market setting. Namely, we consider the case when competing software firms aim at the same application area (e.g. symbolic computation, graphics design, web authoring, etc.) but produce software with features differentiated from each other. Moreover, each consumer would value each of these differentiated features differently.

For example, consider the desktop operating systems provided by Apple Computer and that by Microsoft. Both operating systems are designed to provide an easy-to-use graphical user interface for end users to manage their personal computers. However, some of the features they provide are unique from each other and these differentiated features are valued differently by different camps of consumers.

For simplicity, we model the market as a duopoly and represent the heterogeneity in consumers' preferences for the two different software products using the "linear city" model (Hotelling 1929). In the real software market, very often there would be an innovator firm at the beginning, followed by a number of other software developers who believe it would be profitable to sell similar products in the same application area. However, due to fierce market competition among software developers, usually only a few major players might remain when the market becomes stable. Such a phenomenon of market consolidation can be found in many application areas such as productivity tools, graphics packages, etc. We thus believe that our duopoly model represents an acceptable approximation of reality. We denote the two competing products software 1 (*sw₁*) and software 2 (*sw₂*) respectively.

3.1. The Firms

Before the software is actually developed, the firm would decide on whether to implement any protection mechanism against unauthorized copying. As discussed earlier, a dichotomization of protection strategy should be more appropriate than a continuous level of protection because software products nowadays are usually shipped with protection by simple checking of product key code, or without any protection at all. We denote their decisions on protection strategies by $\eta_i \in \{0,1\}$ ($i = 1,2$) where $\eta_i = 0$ means firm i has chosen not to implement protection. However, if a firm chooses to implement protection ($\eta_i = 1$), an implementation cost k would be incurred.

Once the firms decide on their protection strategies, they also need to decide the selling price of their software p_i ($i = 1,2$). It is assumed that firms would act rationally and

choose their respective optimal prices to maximize their own profits. Assuming a zero marginal cost of production for simplicity, the profit of a firm is given by:

$$\pi_i = p_i d_i - k\eta_i \qquad\qquad \text{….. (1)}$$

where d_i denotes the demand of the software in concern. In order to determine the optimal price, a firm needs to anticipate the possible reactions of both its competitor and the potential consumers, as detailed in the following.

3.2.   The Consumers

We will assume full participation by consumers in that all consumers either buy or pirate one of the software. There are two main factors, namely benefit and cost, affecting the utility a consumer may derive from using a piece of software:

Benefit   The benefit of software would depend on the intrinsic quality of it as well as how its features match the preferences of the user. A piece of software that is highly regarded by one user may not appear to be that useful to another. In our model for horizontally differentiated market, the intrinsic software quality is assumed to be the same for both firms and is denoted by q. On the other hand, consumers are ranked by their preferences of *sw₁* to *sw₂*, with their relative preference positions denoted by $x \in [0,1]$. Namely, a consumer with a smaller x would prefer *sw₁* more to *sw₂*. Considering the deterioration in benefit due to preference mismatch, we model the benefit of *sw₁* to a consumer at position x by $q - tx$ where t denotes the benefit degradation factor (i.e. the "transportation cost") due to preference mismatch. Similarly, the benefit of *sw₂* would be $q - t(1-x)$ to the same consumer.

Cost   The cost would simply be the selling price of the software in case a consumer purchases it. Prices are denoted by p_1 and p_2 for *sw₁* and *sw₂* respectively. In the case of pirating protected software, a cost for pirating would also be incurred. It is assumed that this cost would depend on individual's characteristics (e.g. technical know-how, available resources for pirating, etc.) but would be more or less similar across different target software to be pirated. As such, the consumer's cost for pirating is modeled to be heterogeneous among consumers but the same for both *sw₁* and *sw₂*. For simplicity, we define the cost for pirating as τz where τ is the homogeneous scaling factor of piracy costs while $z \in [0,1]$ denotes the consumer's heterogeneity in pirating protected software. Essentially, some consumers pay a higher cost than others to pirate a piece of protected software.

As a result, the utility U_{b1} and U_{b2} derived from purchasing *sw₁* and *sw₂* would be:

$$U_{b1} = q - tx - p_1 \qquad\qquad \text{….. (2)}$$

$$U_{b2} = q - t(1-x) - p_2 \qquad\qquad \text{….. (3)}$$

As mentioned before, users of pirated software are all subject to risk. We may express this risk factor by the expected benefit of using a pirated software copy. Assuming a risk factor of $\phi \in (0,1)$, the utility U_{p1} and U_{p2} for pirating sw_1 and sw_2 would respectively be:

$$U_{p1} = (1-\phi)(q-tx) - \tau z \eta_1 \qquad\qquad \text{..... (4)}$$

$$U_{p2} = (1-\phi)(q-t(1-x)) - \tau z \eta_2 \qquad\qquad \text{..... (5)}$$

Namely, $1-\phi$ is the probability that the pirated software would work perfectly.

3.3. The Market

Having introduced the essential characteristics of software firms and consumers, we are about to incorporate the two in a competitive market. Namely, we consider a competitive market in a three-stage non-cooperative game. In stage 1, software firms set their respective protection strategies (η_1, η_2). In stage 2, firms determine their optimal prices (p_1, p_2) with the expectation that their profits would be maximized. Lastly in stage 3, consumers choose whether to buy or pirate either one of the software so as to maximize their utilities. The game, in its extensive form, can be illustrated in Figure 1.

We will study the problem using the Sub-game Perfect Nash Equilibrium (SPNE) concept. That is, both firms would first anticipate the utility-maximizing decisions of consumers that would happen at stage 3. Based on this anticipation and also considering the profit-maximizing behavior of its competitor, a firm determines its profit-maximizing price under the 4 possible protection strategies, namely $(\eta_1 = 1, \eta_2 = 1)$, $(\eta_1 = 1, \eta_2 = 0)$, $(\eta_1 = 0, \eta_2 = 1)$ and $(\eta_1 = 0, \eta_2 = 0)$. Lastly, by comparing the maximized profits at each different protection strategy, the two firms set their optimal protection strategies at stage 1 based on the Nash Equilibrium concept. Before proceeding with the analysis, the utility function of the consumer is restated as follows:

$$U = \begin{cases} q - tx - p_1 & \text{if buy } sw_1 \\ (1-\phi)(q-tx) - \tau z \eta_1 & \text{if pirate } sw_1 \\ q - t(1-x) - p_2 & \text{if buy } sw_2 \\ (1-\phi)(q-t(1-x)) - \tau z \eta_2 & \text{if pirate } sw_2 \end{cases} \qquad \text{..... (6)}$$

while the profit functions of the two firms are given by:

$$\begin{aligned} \pi_1 &= p_1 d_1 - k \eta_1 \\ \pi_2 &= p_2 d_2 - k \eta_2 \end{aligned} \qquad\qquad \text{..... (7)}$$
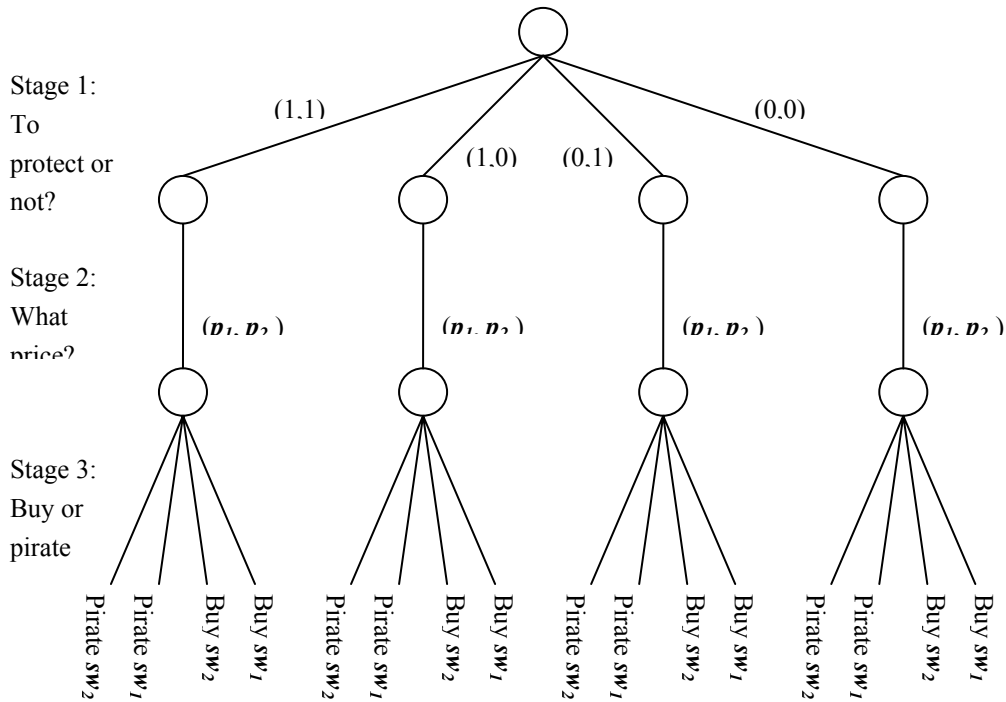
Figure 1: Game Tree

One may easily note from (6) and (7) the dynamics of such a market. Firstly, by setting protection level and price, a firm is affecting the decisions of consumers on whether to buy or pirate sw_1 or sw_2. Such decisions would in turn affect the demands for sw_1 and sw_2. Finally, the profits of firms are actually determined by the demands for their software, as can be seen from (7). To continue our analysis, we will first study the demands for sw_1 and sw_2 that result from the utility-maximizing behavior of consumers.

3.4.   Demand and Profit at Equilibrium

We will consider the demand of sw_1 first. To start with, let us focus on the effect of x on the consumer's utility and leave the effect of z alone for the time being. Figure 2 shows the utility functions for purchasing and for pirating sw_1 at each fixed value of z. Firstly, all utility functions of the consumer are linear in x. For sw_1, the utilities of the two choices (i.e. buying or pirating) would simply be represented by the two downward-sloping straight lines U_{b1} and U_{p1}. In particular, the slope for buying is $-t$ whereas that for pirating is $-(1-\phi)t$. As ϕ must be within the interval $(0,1)$, U_{b1} would always be steeper than U_{p1}.

From Figure 2, it can be easily seen that consumers in the range $(0, x_{b1p1})$ would prefer buying to pirating sw_1 while those in the range $(x_{b1p1}, 1)$ would prefer pirating to buying.

We can now consider the effect of z on the demand. It can be seen from (6) that different values of z would result in different intercepts of the pirating utility, as shown in Figure 3.

Note that the intersection point x_{b1p1} would also be changing with z. Considering z, the indifference point x_{b1p1} becomes a line of indifference between buying and pirating sw_1.

The consumer's utility functions U_{b1} and U_{p1} become 2-dimensional planes in the 3-dimensional space of (U, x, z). The demand for **sw₁** is simply the projection, on the (x, z) plane, of the region where the plane U_{b1} is the highest. Essentially this area is outlined by the indifference line x_{b1p1} with respect to z within the interval $z \in [0,1]$.
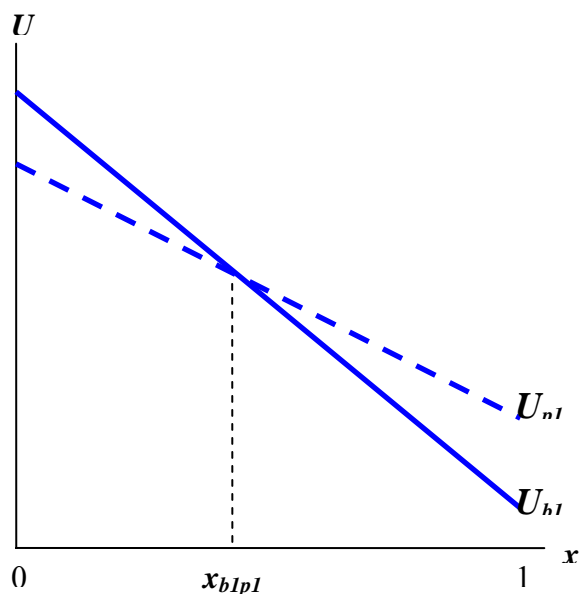

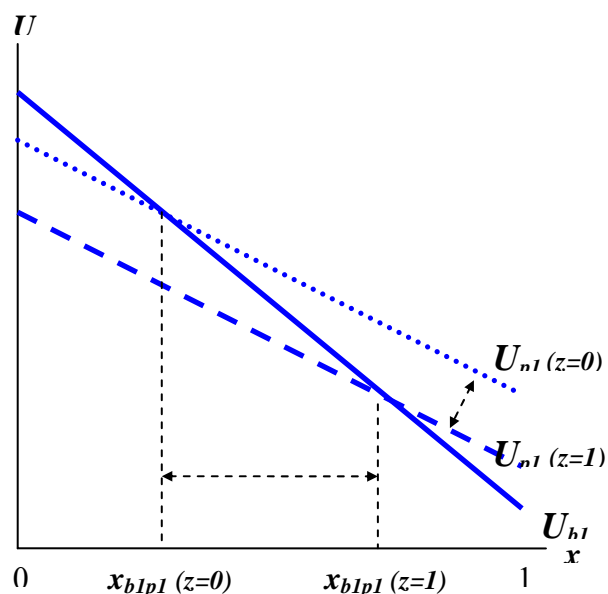Figure 2: Utilities of Buying and Pirating **sw₁**   Figure 3: Effect of **z** on demand

More generally, we can derive the utility-maximizing demands for software from the appropriate lines of indifference, as will be shown later. It should be noted that U_{b1} and U_{p1} may not necessarily intersect within the interval of $x = (0,1)$. This could happen when the vertical intercept of U_{p1} is larger than that of U_{b1}, and in this case pirating would always undercut buying. We would not further analyze this case because the firm will doubtlessly need to lower its price or abandon the market as no consumer is buying the product. On the other hand, U_{b1} and U_{p1} would also not intersect when the value of U_{p1} is smaller than that of U_{b1} when $x = 1$. In this case, buying would always undercut pirating. Again, we would not further analyze this case because there is actually no issue of piracy under such a situation.

We may now consider the situation when **sw₂** is also put in the picture. Considering the 2-dimensional space of (U, x), the utility functions U_{b2} and U_{p2} would simply be two lines having slopes t and $(1 - \phi)t$ respectively, as shown in Figure 4.

We are only interested in the situation when there are non-zero portions of users buying and pirating *sw₁* or *sw₂*. This is equivalent to the following assumption that posits the relations among the intersecting points:

$$1 > x_{b2p2} > x_{p1p2} > x_{b1p1} > 0 \quad \forall \quad z \in [0,1] \qquad \text{..... (C1)}$$

Now we can consider the 3-dimensional space of (U, x, z). As explained previously, the demands for *sw₁* and *sw₂* can be derived from the indifference lines x_{b1p1} and x_{b2p2}. In other words, the demands would simply be the projections on the (x, z) plane as shown in Figure 5. Based on the utility functions of the consumer as given by (6), the demands for *sw₁* and *sw₂* are:

$$d_1 = \int_0^1 x_{b1p1} dz = \frac{1}{2} \frac{\tau \eta_1 - 2p_1 + 2q\phi}{t\phi} \qquad \text{..... (8)}$$

$$d_2 = 1 - \int_0^1 x_{b2p2} dz = 1 + \frac{1}{2} \frac{\tau \eta_2 - 2p_2 + 2q\phi - 2t\phi}{t\phi} \qquad \text{..... (9)}$$
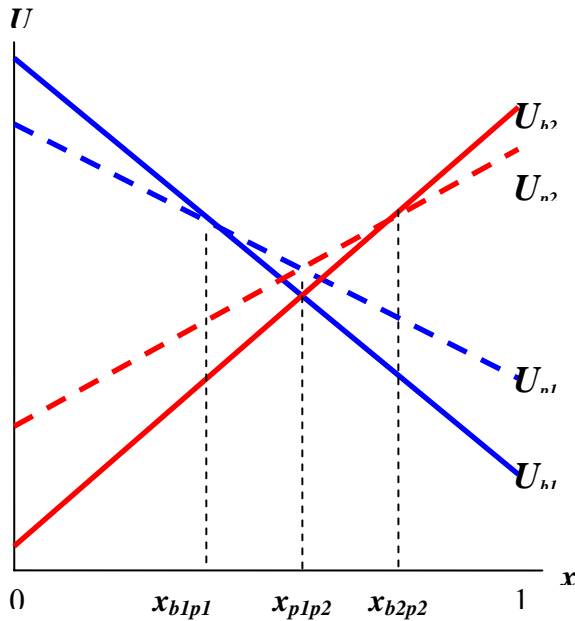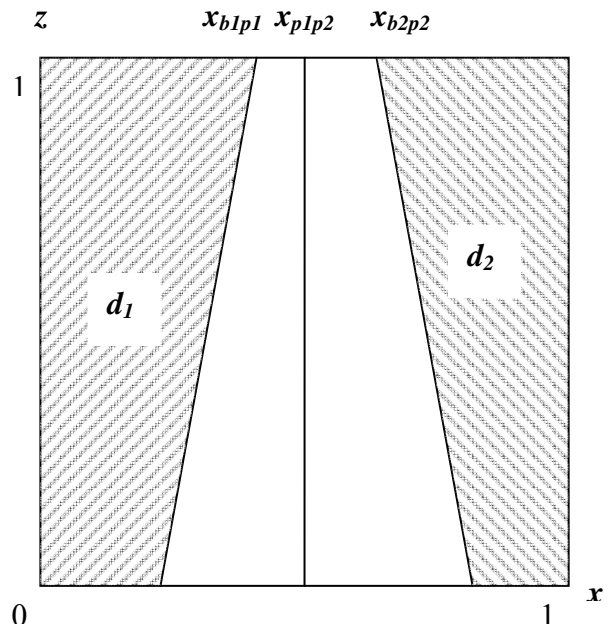

Figure 4:   Utilities of Buying and Pirating
Figure 5:   Demand

That is, the utility-maximizing demands are now entirely in terms of other model parameters, namely the exogenous q, ϕ, t and τ as well as the endogenous p_1, p_2, η_1 and η_2. Similarly, by substituting the anticipated second-stage demands into (7), the firms' profit functions π_1 and π_2 can be expressed entirely in terms of the exogenous q, ϕ, t, τ and k as well as the endogenous p_1, p_2, η_1 and η_2.

Since firms need to determine optimal prices to maximize their profits, their problems would be:

$$\max_{p_1} \pi_1 = \max_{p_1} p_1 d_1 - k\eta_1 = \max_{p_1} \left(\frac{1}{2} \frac{\tau\eta_1 - 2p_1 + 2q\phi}{t\phi} \right) p_1 - k\eta_1 \qquad \ldots\ldots (10)$$

$$\max_{p_2} \pi_2 = \max_{p_2} p_2 d_2 - k\eta_2 = \max_{p_2} \left(1 + \frac{1}{2} \frac{\tau\eta_2 - 2p_2 + 2q\phi - 2t\phi}{t\phi} \right) p_2 - k\eta_2 \qquad \ldots\ldots (11)$$

Proposition 1[4]: In a horizontally differentiated software market, protection would be optimal for both firms only if $k < k^* = \dfrac{\tau(\tau + 4q\phi)}{16t\phi}$**. Otherwise, non-protection would be optimal for both firms.**

Note that k^* is increasing in τ, the pirating cost. Protection would be optimal only if the protection cost is below k^* which mainly consists of a square term of the pirating cost τ of the consumer. This suggests that firms should prefer only a very simple and low-cost protection mechanism or no protection at all.

As shown in the Appendix, the optimal prices, demands and profits of both firms should be symmetrical if they adopt the same protection strategy. In case of protection, the optimal price, demand and profit should be $p_1^* = p_2^* = \dfrac{\tau + 2q\phi}{4}$, $d_1^* = d_2^* = \dfrac{\tau + 2q\phi}{4t\phi}$ and

$$\pi_1^* = \pi_2^* = \frac{\tau^2 + 4\tau q\phi + 4q^2\phi^2 - 16kt\phi}{16t\phi}$$ respectively. In case of non-protection, the optimal

price, demand and profit would be $p_1^* = p_2^* = \dfrac{q\phi}{2}$, $d_1^* = d_2^* = \dfrac{q}{2t}$ and $\pi_1^* = \pi_2^* = \dfrac{q^2\phi}{4t}$

respectively.

It follows that the optimal profit is increasing in q but decreasing in t, regardless of whether protection is in place or not. Namely, optimal profit increases when software is of higher quality but decreases when the negative effect of preference mismatch (i.e. the transportation cost) is high. Also, when protection is in place, the optimal profit is also increasing in τ, the pirating cost on the consumer side. These very much conform to intuition.

Comparing the optimal prices and demands for protection and non-protection, it can be seen that the optimal prices and demands for non-protection would always be lower than those for protection. In this sense, the protection cost contributes as the major determining factor for firms to adopt protection. On the other hand, the effects of risk (i.e. ϕ) on the optimal price, demand as well as profit are summarized in the following table of comparative statics:

---

[4] Proof of Proposition 1 is straightforward and withheld due to length limitation. Interested readers may obtain it from the authors directly.

	$\dfrac{\partial p_1^*}{\partial \phi}$	$\dfrac{\partial p_2^*}{\partial \phi}$	$\dfrac{\partial d_1^*}{\partial \phi}$	$\dfrac{\partial d_2^*}{\partial \phi}$	$\dfrac{\partial \pi_1^*}{\partial \phi}$	$\dfrac{\partial \pi_2^*}{\partial \phi}$
Both Protect	+	+	−	−	+′	+′
Both Not Protect	+	+	0	0	+	+

+′: positive if $\tau < 2q\phi$

In general, optimal prices and profits should increase with risk[5]. It is also interesting to note that the optimal demands would decrease with risk in case of protection. The intuition behind is that if the software is protected and the risk becomes higher, the firm can charge a higher price and reap a higher profit, although there are actually less buying consumers.

4. Vertical Differentiation

Having considered the case of horizontally differentiated market, we now consider the case when software products are vertically differentiated. By vertical differentiation, we mean that software products compete in quality. In general, the higher quality firm can sell at a higher price and also the preference for quality is assumed to be different across different consumers.

Vertical differentiation is indeed quite common in the software industry. Many popular software magazines and download sites (e.g. PC Magazine, Bytes, download.com, zdnet.com, etc.) offer ratings of software products in terms of a common set of criteria such as ease-of-use, functionalities, etc. to assist consumers in making their purchasing decisions.

The main difference from the analysis of horizontal differentiation would be in the formulation of the consumer's utility functions. Namely, we denote the intrinsic qualities of *sw₁* and *sw₂* by q_1 and q_2 respectively. Consumers are now ranked by their preferences to

the quality of software, denoted by $\theta \in (0,1)$, instead of by their preferences of product features (i.e. x). A consumer with a larger θ would derive more utility from the quality of *sw₁* and *sw₂*. As such, the benefits of *sw₁* and *sw₂* to a consumer θ would be $q_1\theta$ and $q_2\theta$

respectively. Without loss of generosity, we assume $q_1 > q_2$. Therefore, the utility functions of the consumer are modified as follows:

$$U = \begin{cases} q_1\theta - p_1 & \text{if buy } sw_1 \\ (1-\phi)q_1\theta - \tau z\eta_1 & \text{if pirate } sw_1 \\ q_2\theta - p_2 & \text{if buy } sw_2 \\ (1-\phi)q_2\theta - \tau z\eta_2 & \text{if pirate } sw_2 \end{cases} \quad \text{….. (12)}$$

where the definitions of $p_1, p_2, \tau, z, \eta_1, \eta_2$ remain the same as in horizontal differentiation.

---

[5] The condition $\tau < 2q\phi$ is likely to be valid because the pirating cost should be significantly lower than the benefit of the

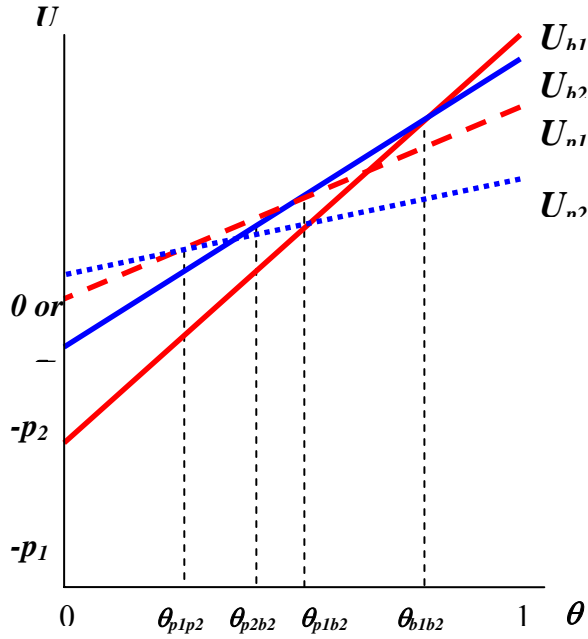software to make a case of software piracy.
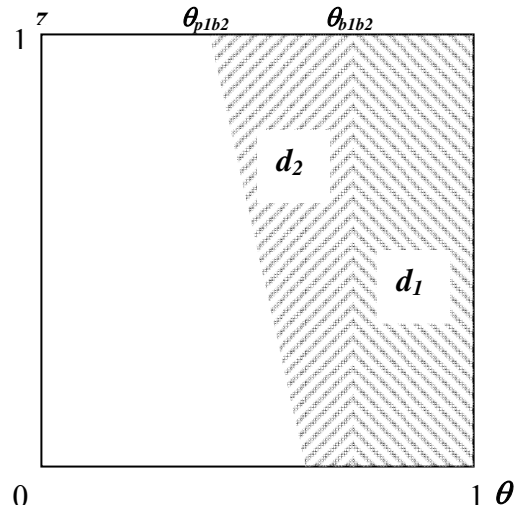
Figure 6: Consumer's Utility

Figure 7: Demands

Consider the consumer's utility when pirating either *sw₁* or *sw₂*. From (12), it follows that the slope of consumer's utility with respect to θ when pirating *sw₁* would always be steeper than that of pirating *sw₂*. It can be easily seen that when (η_1, η_2) is either $(1,1)$, $(0,1)$ or $(0,0)$, pirating *sw₁* would always dominate pirating *sw₂*. When $(\eta_1, \eta_2) = (1,0)$, pirating *sw₂* may dominate pirating *sw₁* for some consumers. Denote the point of indifference between pirating *sw₁* and *sw₂* as θ_{p1p2}, it can be seen that no one would pirate *sw₂* unless $\theta_{p1p2} > 0$.

Now consider the consumer's utility when buying either software. From (12), it is clear that the slope of consumer's utility with respect to θ when buying *sw₁* would always be steeper than that of buying *sw₂*. We assume that the point of indifference between buying *sw₁* and buying *sw₂*, denoted by θ_{b1b2}, lies somewhere between 0 and 1 (i.e. $0 < \theta_{b1b2} < 1$). (Without this assumption, buying either software would always dominate buying the other.)

Consider the consumer's utility with respect to θ as shown in Figure 6. Namely, there will be 4 groups of consumers. Those with $\theta \in [\theta_{b1b2}, 1]$ would buy *sw₁* and those with $\theta \in [\theta_{p1b2}, \theta_{b1b2}]$ would buy *sw₂*. Moreover, those with $\theta \in [\theta_{p1p2}, \theta_{p1b2}]$ would choose to pirate *sw₁* while those with $\theta \in [0, \theta_{p1p2}]$ (if $(\eta_1, \eta_2) = (1,0)$ and $\theta_{p1p2} > 0$) would choose to pirate *sw₂*. However, we assume that even though *sw₂* is unprotected, it can only convert a certain portion of consumers from pirating *sw₁* to pirating *sw₂*, mainly because the pirating cost should be relatively small. More formally, we assume the relations among the intersecting points to satisfy the following:

$$1 > \theta_{b1b2} > \theta_{p1b2} > \theta_{p2b2} \quad \forall \quad z \in [0,1] \qquad \qquad \text{….. (C2)}$$

133

Intuitively, it implies that the price of *sw₁* is higher than that of *sw₂* and both prices are higher than the pirating cost. Also, the risk of using pirated software is more significant than the quality difference between the products (i.e. $q_2 > (1-\phi)q_1$ such that U_{b2} is always steeper than U_{p2}).

Similar to our analysis on the horizontal differentiation case, the demands are given by the corresponding projections of the utility planes onto the (, *z*) plane, as illustrated by the shaded areas in Figure 7. Namely, the demands for *sw₁* and *sw₂* are given by:

$$d_1 = 1 - \int_0^1 \theta_{b1b2} dz = 1 - \frac{p_1 - p_2}{q_1 - q_2} \qquad \dots (13)$$

$$d_2 = \int_0^1 \theta_{b1b2} dz - \int_0^1 \theta_{p1b2} dz = \frac{p_1 - p_2}{q_1 - q_2} + \frac{1}{2}\frac{\tau\eta_1 - 2p_2}{q_1(\phi-1)+q_2} \qquad \dots (14)$$

That is, the utility-maximizing demands would be entirely in terms of the other exogenous model parameters q_1, q_2, ϕ, τ and k as well as the endogenous p_1, p_2, η_1 and η_2. Since firms need to determine optimal prices to maximize their profits, their problems would be:

$$\max_{p_1} \pi_1 = \max_{p_1} p_1 d_1 - k\eta_1 = \max_{p_1}\left(1 - \frac{p_1 - p_2}{q_1 - q_2}\right)p_1 - k\eta_1 \qquad \dots (15)$$

$$\max_{p_2} \pi_2 = \max_{p_2} p_2 d_2 - k\eta_2 = \max_{p_2}\left(\frac{p_1 - p_2}{q_1 - q_2} + \frac{1}{2}\frac{\tau\eta_1 - 2p_2}{q_1(\phi-1)+q_2}\right)p_2 - k\eta_2 \qquad \dots (16)$$

Proposition 2[6]: In a vertically differentiated software market, non-protection would always be optimal to the lower-quality firm. Protection would be optimal for the higher-quality firm only if $k < k^* = \dfrac{\tau(8q_1^2\phi + q_1\tau - 8q_1q_2\phi - \tau q_2)}{4(q_1 + 3q_1\phi - q_2)^2}$ **.**

It is apparent that k^* is always increasing in τ, the pirating cost. Protection would be optimal only if the protection cost is below k^* which mainly consists of a square term of the pirating cost τ of the consumer. This once again suggests that firms should prefer only a very simple and low-cost protection mechanism or no protection at all.

As shown in the Appendix, the optimal prices, demands as well as profits, if non-protection strategy is adopted by both firms, would be:

$$p_1^* = \frac{2q_1\phi(q_1 - q_2)}{q_1 + 3q_1\phi - q_2} \qquad d_1^* = \frac{2q_1\phi}{q_1 + 3q_1\phi - q_2} \qquad \pi_1^* = \frac{4q_1^2\phi^2(q_1 - q_2)}{(q_1 + 3q_1\phi - q_2)^2}$$

$$p_2^* = \frac{(q_1\phi - q_1 + q_2)(q_1 - q_2)}{q_1 + 3q_1\phi - q_2} \qquad d_2^* = \frac{q_1\phi}{q_1 + 3q_1\phi - q_2} \qquad \pi_2^* = \frac{q_1\phi(q_1\phi - q_1 + q_2)(q_1 - q_2)}{(q_1 + 3q_1\phi - q_2)^2}$$

---

[6] Proof of Proposition 2 is straightforward and withheld due to length limitation. Interested readers may obtain it from the authors directly.

In particular, the higher-quality firm would charge a higher price and the optimal profit of the higher-quality firm is increasing in q_1 just like the case of horizontal differentiation. The optimal profit of the lower-quality firm is also increasing in ϕ. In case the higher-quality firm chooses to protect, the optimal prices, demands as well as profits of the two firms would become:

$$p_1^* = \frac{(4q_1\phi + \tau)(q_1 - q_2)}{2(q_1 + 3q_1\phi - q_2)} \qquad\qquad d_1^* = \frac{1}{2}\frac{4q_1\phi + \tau}{q_1 + 3q_1\phi - q_2}$$

$$\pi_1^* = \frac{1}{(q_1 + 3q_1\phi - q_2)^2}(16q_1^3\phi^2 + 8\tau q_1^2\phi + q_1\tau^2 - 16q_1q_2\phi^2 - 8q_1q_2\phi\tau - q_2\tau^2 - 4kq_1^2 - 24kq_1^2\phi + 8kq_1q_2 - 36kq_1^2\phi^2 + 24kq_1q_2\phi - 4kq_2^2)$$

$$p_2^* = \frac{(q_1\phi - q_1 + q_2 + \tau)(q_1 - q_2)}{q_1 + 3q_1\phi - q_2} \qquad d_2^* = \frac{q_1\phi(q_1\phi - q_1 + \tau + q_2)}{(q_1\phi - q_1 + q_2)(q_1 + 3q_1\phi - q_2)}$$

$$\pi_2^* = \frac{q_1\phi(q_1\phi - q_1 + \tau + q_2)^2(q_1 - q_2)}{(q_1\phi - q_1 + q_2)(q_1 + 3q_1\phi - q_2)^2}$$

       Similar to the case of horizontal differentiation, it can be shown that the optimal prices and demands for non-protection would always be lower than those if the higher quality firm chooses to protect. Again, the protection cost should be the major determining factor for the higher quality firm to choose protection. The effects of risk on the optimal price, demand as well as profit can be seen from the following table of comparative statics:

	$\dfrac{\partial p_1^*}{\partial \phi}$	$\dfrac{\partial p_2^*}{\partial \phi}$	$\dfrac{\partial d_1^*}{\partial \phi}$	$\dfrac{\partial d_2^*}{\partial \phi}$	$\dfrac{\partial \pi_1^*}{\partial \phi}$	$\dfrac{\partial \pi_2^*}{\partial \phi}$
Only Firm 1 Protect	$+''$	$+''$	$+''$	$-''$	$+''$	$+''$
Both Not Protect	$+$	$+$	$+$	$+$	$+$	$+$

$+''$: positive if $\tau < \dfrac{4(q_1 - q_2)}{3}$ $-''$: negative if $\tau < \dfrac{4(q_1 - q_2)}{3}$

Namely, optimal prices, demands and profits would all increase with risk if both firms choose the non-protection strategy. In case the higher quality firm chooses to protect, the optimal prices and profits would still increase with risk[7] but the optimal demand for the lower quality product would decrease. Intuitively, a higher risk would lead to a higher demand for the higher quality product but that would essentially reduce the demand of the lower quality product, although it would still result in an increased profit for the lower quality firm due to the increased price.

5. Discussion

       We have considered the optimal protection strategies under two different types of market differentiations. Our analysis shows that in a vertically differentiated market the lower-quality firm would always prefer non-protection. Furthermore, protection would be

---

[7] We mainly consider the case when $\tau < (4/3)(q_1 - q_2)$ because the pirating cost should be significantly lower than the benefits of software in order for people to consider pirating.

profit-maximizing only if the implementation cost is low. In reality, we observe that most software developers are still implementing some sort of protection mechanisms in their software products although strong externalities effects are quite universal in the widely-expanded software business nowadays. Apparently this seems to be inconsistent with the findings from previous researches (Conner & Rumelt 1991; Shy & Thisse 1999). It is also interesting to note that complicated protection mechanisms are seldom found nowadays. Most software products just employ a simple product key code validation or registration process to guard against unauthorized usage. The duplicated uses of those product key codes are seldom really checked, nor would they prevent the actual usage of the software. The software protection mechanisms used nowadays are observed to be converging to a very simple and common form of product key code validation. It is reasonable to expect that this sort of mechanisms would not be very costly to implement and is likely to be reusable on other software products developed by the same firm, further lowering the average implementation cost of protection.

Anecdotal evidence supports some of our findings. For instance, Maplesoft's Maple and Wolfram's Mathematica can be regarded as two horizontally differentiated products competing in the area of symbolic computation software. They are two incompatible products with different sets of features and employ different file formats although their problem domains are similar. Both of them adopt a simple product key code validation mechanism for protection. It is interesting to find that apart from protection strategy, their listed prices are also very similar. These facts are consistent with our analytical findings.

On the other hand, the Microsoft Office suite and Sun Microsystems' StarOffice package can be regarded as examples of vertically differentiated products. As Microsoft Office has become a de facto standard in the office productivity area, competitors in this application area have to develop products that are compatible with it. Unfortunately, Microsoft Office has been adopting a proprietary file format that prevents others from producing compatible products. StarOffice actually originated from an open-source initiative that attempted to reverse-engineer the proprietary file formats used by Microsoft Office that was believed to be over-priced. As a result, StarOffice is able to read and produce documents and spreadsheets in Microsoft Office format. However, it never achieves full compatibility with Microsoft Office and users may perceive it as a lower-quality product in this sense. Interestingly, Microsoft Office is adopting a protection scheme while StarOffice is not. Again, this is consistent with our analytical findings.

6.   Conclusion and Future Work

Our findings show that the primary consideration of software protection strategies should be the implementation cost. This may help explain why complicated protection mechanisms have mostly been driven out of the software market nowadays. Also, the software developer of the lower-quality product would tend to adopt non-protection in order to compete with the higher-quality product in a vertically differentiated market. We have also considered the effects of risk in using pirated software. In either horizontally differentiated or vertically differentiated markets, a firm's profit is found to be increasing with the risk of using pirated software[8]. More interestingly, one may find that risk has a similar positive effect on profit as product quality.

As mentioned in the beginning of this paper, the existence of risk in itself does not prevent the use of pirated software. However, consumers are deterred from doing so in view

---

[8] If protection is chosen, this is subject to some technical constraints regarding the pirating cost τ.

of the uncertainty introduced by the risk. Gopal & Sanders (1997) showed that deterrent measures would be more profit-maximizing than preventive ones for dealing with software piracy. Our findings about the effect of risk also support this argument. We attribute one of the sources of the risk in using pirated software to the unavailability of technical support. However, it should be noted that in this study we do not assume risk to be a parameter controllable by firm. In real life, the risk in using pirated software is subject to many other factors beyond the control of the software firm. We believe that firms can actually affect at least two factors: the reliability of software product and the quality of technical support. Ironically, the value of technical support increases when the reliability of software decreases. It would be interesting to see how these two factors (i.e. reliability and technical support quality) would interact and affect the firm's optimal strategy. We believe that future extension of this study can be developed along this line.

Our analysis does not consider the effect of externalities. Intuitively, firms should merely have less incentive to protect their software if there are strong externalities effects. However, empirical observation shows that most software products are still implementing protection mechanisms, albeit only simple and low-cost ones. In this study, we mainly focus on the effects of protection costs and risk in a competitive market. It would actually be straightforward to extend our model to include externalities effects as well. This could be another possible extension of this work.

7.　References

Banerjee, D.S. (2003) "Software Piracy: A Strategic Analysis and Policy Instruments," *International Journal of Industrial Organization (21)*, 2003, pp. 97-127.

Business Software Alliance (2003) *Eighth Annual BSA Global Software Piracy Study: Trends in Software Piracy 1994-2002*, 2003.

Chen, Y.N. and Png, I. (2003) "Information Goods Pricing and Copyright Enforcement: Welfare Analysis," *Information Systems Research (14:1)*, March 2003, pp. 107-123.

Cheng, H.K., Sims, R.R. and Teegen, H. (1997) "To Purchase or to Pirate Software: An Empirical Study," *Journal of Management Information Systems (13:4)*, Spring 1997, pp. 49-60.

Conner, K.R. and Rumelt, R.P. (1991) "Software Piracy: An Analysis of Protection Strategies," *Management Science (37:2)*, February 1991, pp. 125-139.

Gopal, R.D. and Sanders, G.L. (1997) "Preventive and Deterrent Controls for Software Piracy," *Journal of Management Information Systems (13:4)*, Spring 1997, pp. 29-47.

Gopal, R.D. and Sanders, G.L. (1998) "International Software Piracy: Analysis of Key Issues and Impacts," *Information Systems Research (9:4)*, December 1998, pp. 380-396.

Hotelling, H. (1929) "Stability in Competition," *The Economic Journal*, March 1929, pp.41-57.

Shy, O. and Thisse, J.F. (1999) "A Strategic Approach to Software Protection," *Journal of Economics & Management Strategy (8:2)*, Summer 1999, pp. 163-190.

Tirole, J. (1988) *The Theory of Industrial Organization*, 1988.

US-CERT (2004) *US-CERT Vulnerability Notes* (http://www.kb.cert.org/vuls), 2004.