ICIS 2005 Proceedings

International Conference on Information Systems (ICIS)

December 2005

An Economic Analysis of the Software Market with a Risk-Sharing Contract

Byung-Cho Kim
Carnegie Mellon University

Pei-Yu Chen
Carnegie Mellon University

Tridas Mukhopadhyay
Carnegie Mellon University

An Economic Analysis of the Software Market with a Risk-Sharing Contract

Byung Cho Kim, Pei-yu Chen, and Tridas Mukhopadhyay
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA U.S.A.
bckim@andrew.cmu.edu **pychen@andrew.cmu.edu**
tridas@andrew.cmu.edu

Abstract

Low quality of software has been blamed for poor security of our computer networks as major viruses and worms exploit the vulnerabilities of such software. However, software vendors have no incentive to improve the quality of their products since they are not directly liable for any loss due to poor quality. Software liability has been intensely discussed among computer scientists and jurists for years as a possible solution for software quality improvement. This paper proposes a risk-sharing mechanism between software vendors and customers as a market-driven method to impose software liability. We consider two dimensions of software quality: functionality and security quality. We present an economic model of the software market with a risk-sharing mechanism, which takes into account the strategic interplay of risk-sharing and security quality of the software given a certain level of functionality. We then apply this model in different scenarios, and examine the implications of the risk-sharing mechanism in the context of cyber security. Our model provides evidence of under-provided security quality of software in the monopoly case, as has been observed in the market. We consider the feasibility and effectiveness of the risk-sharing mechanism under various scenarios, and find the conditions under which the proposed mechanism is promising.

Keywords: Cyber security, software quality, risk-sharing

Introduction

As the Internet has revolutionized the way individuals, industry, and the government communicate and conduct their daily business, the intensive interconnectivity has increased the vulnerability of computer systems. Consequently, network security becomes a major issue for electronic business and corporate communications. To cope with the new risk, the computer industry has tried to develop new weapons such as firewalls, encryption techniques, access control mechanisms, and intrusion detection systems. The federal government has formed the Department of Homeland Security and is developing a national strategy to secure cyber space. Despite these efforts, the security level of computer networks is still low, and the potential loss is enormous. Ernst & Young's Global Information Security Survey 2004 shows that only 20 percent of the respondents strongly agreed that their organizations perceive information security as a CEO-level priority and that less than half agreed that they could continue business operations in the event of a serious disruption. The Computer Security Institute and the Federal Bureau of Investigation reported that total losses due to computer crimes and computer security incidents for the respondents to the 2004 survey were over 141 million U.S. dollars (Gordon et al. 2004).

Fisk (2002) argues that there are well known technical and procedural techniques for preventing computer system vulnerability. However, applying these techniques can be resource-intensive and will not be done without a sufficient incentive. One major reason for poor security is that the software industry is at a suboptimal, but self-supporting equilibrium that does not support the efforts required for software security improvement. Customers do not have good enough safeguards, both because available options on the security market seem to be ineffective and too expensive, and the value of running safe operations is not fully appreciated. They have learned to tolerate low-quality software, enabling the vendors to be successful without improving the quality of their products. On the vendor side, both a perceived small market and high development costs have made producing highly secure software a significant risk.

Slow growth of the security market and low quality of software have been identified as main causes of the poor state of network security (Yurcik and Doss 2002). However, software vendors have no incentive to improve the quality of their products since they are not directly liable for any loss due to poor quality. To solve this problem, security experts suggest legal liability and cyber insurance as possible solutions (Schneier 2004; Varian 2000). Unfortunately, not much research has been done on this problem from an economic perspective. According to Fisher (2002), some companies are already demanding liability clauses in contracts with vendors, holding the vendors responsible for any security breach connected to their software, quoting Karl Keller, president of IS Power Inc., who says, "Contractual liability is a great motivator. I'm encouraged that liability for vulnerabilities is entering into contracts."

In this paper, we propose a risk-sharing mechanism between software vendors and customers as an alternative solution to improve software quality. We present an economic model of the software market, which takes into account the strategic interplay of risk-sharing and software quality. Our model considers two dimensions of software quality, which are functionality and security quality. Our focus throughout the paper is security quality. We first examine the implications of the risk-sharing mechanism both in the monopoly and socially optimal cases. We find evidence of under-provided quality of software under monopoly, as has been observed in the market. The results show that the social planner who maximizes social surplus offers higher-quality product than the monopolist and that risk-sharing and security quality are not strategic complements. This intuition provides insights for the otherwise unexpected result that neither the social planner nor the monopolist has any incentive to bear the risk. This is interesting in the sense that even for the social planner, sharing risk with the customers is not optimal at equilibrium, although risk-sharing mechanisms such as warranties are widely used in other industries.

We extend the model to a duopoly competition. We start by examining the case where the entrant brings a product with the same quality level as the incumbent who does not want to share any risk. Unlike the monopoly case, we find that the entrant has an incentive to introduce positive risk-sharing to alleviate competition and that the risk-sharing level increases as the quality level increases. Then we extend this scenario to the case where two vendors with products of the same quality differentiate their products by offering different levels of risk-sharing. We find that in the presence of competition, where both vendors differentiate their products not by quality but by risk-sharing, the high-value vendor is willing to share the risk whereas sharing no risk is the optimal choice for the low-value vendor. The high-value vendor's optimal level of risk-sharing is the same as the risk-sharing level of the entrant in the first duopoly scenario, which increases as the quality increases.

Model

We analyze a software vendor's decision on the quality and the risk-sharing levels, using a model of vertical quality differentiation (Mussa and Rosen 1978; Ronnen 1991). There are two types of players in the market: a software vendor and customers. It is shown that the best strategy for the software vendors is to introduce their products as early as possible and then to patch them later (Arora et al. 2005). Consequently, the initial quality of software products is low. Security experts argue that the defects of such software are exploited by malicious hackers to attack computer systems, and that the quality of the general software in terms of security should be improved. Customers in our model are considered to be firms that are likely to have higher incentive to adopt security solutions than do individual consumers, whose awareness of security in general is still low in reality. We assume that increasing the level of security quality of software reduces the expected loss from cyber attack in the life-span of the product. This is reasonable in the sense that attacks on computers or systems with more secure software are less likely to be successful.

Customer's Utility Function

We consider two dimensions of software quality: functionality and security quality. In early 2002, Microsoft stopped all Windows feature development, and focused only on security improvement. Our model considers a vendor that emphasizes security quality given a certain level of functionality ($V > 0$). Let q be the security quality of the software product where $q \in [0, 1]$. Security quality measures vulnerability of the software to attacks at the product launch. Bug-free software can be considered to be of perfect security quality. Following the argument of security experts that the initial quality of the software at the product launch matters, and that the availability of patching mechanism may worsen the situation, we focus on the initial quality in our model. Let $K(q)$ be the expected loss in the product life-span when q-quality software is installed. Under the proposed risk-sharing mechanism, the vendor takes some proportion of the risk, denoted by r where $r \in [0, 1]$. If any attack exploiting the vulnerability of the installed software is successful and incurs loss, then the vendor shares the responsibility with its customers. Thus, the expected utility of a customer who purchases the software with price p is

$$E(U) = \theta[V - (1 - r)K(q)] - p$$

$K(q)$ in our model is based on a certain period of time. It is reasonable since most software products are licensed to the corporate customers. Thus, the life-span of the software is considered to be the licensing period. We assume that $K'(q) < 0$ and $K''(q) > 0$, so that the expected loss decreases as the quality level increases at a diminishing rate. θ captures customer heterogeneity, indicating how much utility a customer derives from the software's functionality. The same attack may cause more severe damage to some firms than others. If θ is high, the customer is more sensitive to security features of the product, in that she enjoys more utility from the product, but also suffers more disutility from a successful attack. It holds in reality that some firms are more sensitive to security than others. For example, banks may be such customers with high θ. We assume that θ is uniformly distributed on [0, 1]. Customers who have expected utility greater than zero buy the software, whereas others do not.

Vendor's Profit Function

The software vendor's expected profit is

$$E(\pi(p, q, r)) = D(p, q, r)(p - rK(q)) - C(q)$$

where $D(p, q, r)$ is the demand for the product, p is the price, and $C(q)$ represents the fixed cost for producing software with quality level q. Production of information goods such as software involves high fixed cost but low variable cost. In other words, the cost of producing the original copy is substantial whereas the cost of producing additional copies is negligible. As a result, given the context of software product, the cost does not depend on quantity, meaning that the variable cost of production is zero. We assume a convex cost function, that is, $C'(q) > 0$ and $C''(q) > 0$, so that the cost increases as the quality level rises at a growing rate. $rK(q)$ is the expected loss, for which the vendor is responsible per unit of the product. Although the variable cost of production is assumed to be zero, $rK(q)$ plays a role of the variable cost in our model.

Monopoly Versus Social Optimum

The monopoly case is quite relevant to software industry. Consider the case of Microsoft, which dominates the PC operating systems market. We analyze a three-stage game. At the first stage, the monopolistic vendor decides the quality level q and the risk-sharing level r simultaneously, and at the second, the vendor sets up the price p. Then the customers decide whether or not to buy the product at the last stage.

> **Proposition 1**: *In a software market with a risk-sharing mechanism, neither the monopolist nor the social planner is willing to share any risk. At equilibrium, the social planner offers a higher-quality product than the monopolist.*[1]

This is interesting in the sense that neither the social planner nor the monopolist has any incentive to bear the risk. Note that the risk-sharing factor does not affect the fixed cost and that sharing no risk allows the social planner to face zero marginal cost and to cover the entire market. The social planner is left with no resource to share the loss when it serves the entire market by offering a price at marginal cost. Thus, it turns out that even the social planner does not want to share any risk. Interestingly, the risk-sharing factor and the quality in our model are not strategic complements. In other words, it is not always true that the factors that increase the risk-sharing level also result in higher quality. Proposition 1 provides evidence of under-provided quality of software under monopoly, as has been observed in the market. Figure 1 illustrates the relationship between the quality of a monopolist and a social planner.

[1] The mathematical proofs of all the propositions can be obtained from the authors upon request.

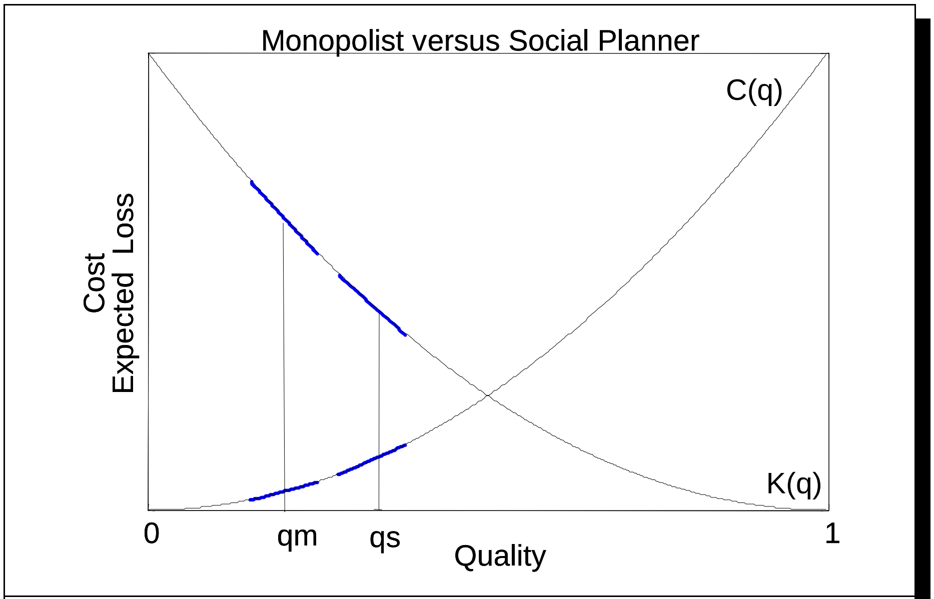

Figure 1. Equilibrium Quality of the Monopolist and the Social Planner

Competition

Incumbent and Entrant with Same Quality

We now study the case of a duopoly market with an incumbent and an entrant offering software products of the same quality. This scenario captures the market where there are a monopolistic incumbent that has no incentive to share the risk and an entrant that enters the market bringing a product of the same quality as the incumbent's product, that is, $q_E = q_I = \overline{q}$. Let $C_E(q)$ be the entrant's cost for developing *q*-quality software and $C_I(q)$ be the same for the incumbent. In reality, software development costs are declining, meaning that the entrant faces a lower development cost than the incumbent. Thus, we assume that $C_E(q) < C_I(q)$ for any level of *q*. We investigate whether the entrant has an incentive to share the risk and, if so, how much it will share at equilibrium. In this game, the entrant chooses its optimal risk-sharing level first. Then both the incumbent and the entrant set up the price simultaneously. At the last stage, customers decide whether to buy from the incumbent or the entrant or neither.

> **Proposition 2**: *In the presence of competition, the entrant offering the same quality software as the incumbent sharing no risk has an incentive to introduce positive risk-sharing to alleviate competition. Moreover, as the quality level increases, the risk-sharing level also increases.*

In contrast to the monopolist and the social planner, the entrant in this scenario has an incentive to share the risk. It follows that the risk-sharing level increases as the quality level increases. This result is quite interesting in the sense that without the risk-sharing mechanism, the entrant may have less incentive to enter the market because its entry may trigger Bertrand-like price competition.

Duopoly Competition with Same Quality but Different Risk-Sharing

This scenario serves as an extended case to the previous one. In this scenario, two vendors compete against each other with the product of the same quality and they differentiate their products by offering different levels of risk-sharing. We label the vendor sharing high risk, hence offering high value to customers, as H vendor and denote the other vendor sharing low risk, hence offering low value to customers, as L vendor.

Proposition 3: *In the presence of competition where two vendors offer same-quality software, at equilibrium, risk-sharing acts as a differentiator so that one firm shares positive risk,* $r_H^* = \frac{3(V - K(\overline{q}))}{4K(\overline{q})}$ *, and thus offers higher value to customers, while sharing no risk is the optimal choice for the other firm* ($r_L^* = 0$).

Proposition 3 shows the optimal risk-sharing level for both the high-value and the low-value vendor. When they differentiate their products by offering different levels of risk-sharing, the high-value vendor has an incentive to share positive risk whereas the low-value vendor is not willing to bear any risk. This is interesting in the sense that for the low-value vendor, risk-sharing may seem to be risky when it perceives that its rival will share higher risk than itself. Interestingly, we find that the high-value vendor's optimal risk-sharing level increases as the quality level increases. Thus, customers may use the high-value vendor's risk-sharing level as a proxy of the quality level given in the market.

Conclusion

To enhance the poor state of network security, one needs to solve the fundamental problem: giving software vendors an incentive to increase the quality of their products. As a possible solution, software liability has been discussed for years among computer scientists, jurists, and policy makers. In this paper, we propose a risk-sharing mechanism as a market-driven method to impose software liability. We present an economic model of the software market, which takes into account the strategic interplay of risk-sharing and quality of product. Our research contributes to the literature in the following ways. First, we provide an economic framework for analyzing a security issue where little research has dealt with the problem of the software market from an economic perspective although the solution to this problem is economic rather than technical. Second, our results suggest that a risk-sharing mechanism is promising under certain conditions as a form of market-driven regulation. Finally, our results have managerial implications for software vendors. Especially, a vendor who enters the market competing with an established incumbent may want to consider sharing the risk with its customers.

While significant, this study can be improved in several ways. First, examining how different forms of government policy affect software quality will be an interesting topic for future research. Proposals for government action by policy makers include offering tax incentives to businesses for spending on security. It will be meaningful to investigate whether the government's subsidizing policy is better than the vendor-side regulation in terms of software quality improvement. Second, developing a way to apply the risk-sharing mechanism may form a separate research area. For example, loss measurement and risk analysis are prerequisites for the proposed risk-sharing mechanism. Finally, considering a mechanism for patching and examining how the availability of patching affects the interplay of risk-sharing and software quality will be interesting.

References

Arora, A., Calkins, J., and Telang, R. "Sell First, Fix Later: Impact of Patching on Software Quality," *Management Science*, 2005, forthcoming.

Ernst & Young. "Global Information Security Survey 2004," 2004 (available online at **http://www.ey.com/global/download.nsf/International/2004_Global_Information_Security_Survey/$file/2004_Global_Information_Security_Survey_2004.pdf**).

Fisher, D. "Contracts Getting Tough on Security," *eWeek*, April 15, 2002.

Fisk, M. "Causes and Remedies for Social Acceptance of Network Insecurity," paper presented at the Workshop on Economics and Information Security, University of California, Berkeley, May 16-17, 2002 (available online at **http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/econws/35.pdf**).

Gordon, L. A., Loeb, M. P., Lucyshyn, W., and Richardson, R. "2004 CSI/FBI Computer Crime and Security Survey," Computer Security Institute, 2004 (available online at **http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf**).

Mussa, M., and Rosen, S. "Monopoly and Product Quality," *Journal of Economic Theory* (18), 1978, pp. 301-317.

Ronnen, U. "Minimum Quality Standards, Fixed Costs, and Competition," *The RAND Journal of Economics* (22), 1991, pp. 490-504.

Schneier, B. "Information Security: How Liable Should Vendors Be?," *Computer World*, October 28, 2004 (available online at **http://www.schneier.com/essay-073.html**).

Varian, H. R. "Managing Online Security Risks," *New York Times*, June 1, 2000 (available online at **http://www.nytimes.com/library/financial/columns/060100econ-scene.html**).

Yurcik, W., and Doss, D. "Cyberinsurance: A Market Solution to the Internet Security Market Failure," paper presented at the Workshop on Economics and Information Security, University of California, Berkeley, May 16-17, 2002 (**http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/econws/53.pdf**).