Software Security and Liability

Byung-Cho Kim  
*Carnegie Mellon University*

Pei-Yu Chen  
*Carnegie Mellon University*

Tridas Mukhopadhyay  
*Carnegie Mellon University*, tridas@andrew.cmu.edu

Follow this and additional works at: [http://repository.cmu.edu/tepper](http://repository.cmu.edu/tepper)  
Part of the *Economic Policy Commons*, and the *Industrial Organization Commons*
Software Security and Liability

Byung Cho Kim  
Tepper School of Business  
Carnegie Mellon University  
Pittsburgh, PA 15213  
bckim@andrew.cmu.edu

Pei-yu Chen  
Tepper School of Business  
Carnegie Mellon University  
Pittsburgh, PA 15213  
pychen@andrew.cmu.edu

Tridas Mukhopadhyay  
Tepper School of Business  
Carnegie Mellon University  
Pittsburgh, PA 15213  
tridas@andrew.cmu.edu

December 25, 2005
Abstract

The abundance of flawed software has been identified as the main cause of the poor security of computer networks since major viruses and worms have been exploiting the vulnerabilities of such software. As an incentive mechanism for software security quality improvement, software liability has been intensely discussed among computer scientists, jurists, and policy makers for a long time. In this paper, we examine how the liability mechanism affects a monopolistic software vendor’s decision on security quality and market coverage. We then analyze the welfare implications of the liability mechanism. We find that high marginal willingness to pay for the software leads to full market coverage without liability. When liability is imposed, full market coverage obtains only if the expected loss is bounded. We also find that security quality is underprovided without liability while socially optimal level is offered with liability. Interestingly, our results indicate that imposing liability may discourage the monopolist from improving security while it leads to higher consumer surplus. When the marginal willingness to pay is relatively low, the liability mechanism brings higher social surplus. In the presence of information asymmetry between the vendor and the customers, the liability mechanism yields higher security quality and higher consumer surplus.
1 Introduction

Software has become an indispensable part of our daily lives. Not only large-scale operating systems in businesses but also home electric appliances such as television sets and telephones are powered by thousands of program instructions. As our society depends more on software, its malfunction becomes more disastrous. For example, the task force investigating the cause of the electricity blackout on August 14, 2003, which crippled much of the Northeast USA and parts of Canada concluded that a software failure at Akron, Ohio-based FirstEnergy Corporation might have contributed significantly to the power outage (Verton 2003). The explosion of software usage has increased the vulnerability of computer systems, highlighting security concerns.

According to Microsoft Progress Report on Security, malicious software code has been around for decades, but only in the last few years, the Internet, high-speed connections and millions of new computing devices have converged to create a truly global computing network in which a virus or worm can circle the world in a matter of minutes. Consequently, external attacks such as viruses and worms that exploit the vulnerabilities of the operating systems get more serious. Ernst & Young’s Global Information Security Survey 2004 found that 77% of the respondents rated major viruses and worms to have the highest threat intensity (Ernst & Young 2004).

The abundance of flawed software has been identified as the main cause of the poor security of computer networks (Yurcik and Doss 2002). In September 2003, Steven Adler, senior security strategist for Microsoft Corporation apologized for the damage and losses caused by the onslaught of computer viruses that have attacked his company’s software. According to Fisk (2002), the software industry is at a sub-optimal, but self-supporting equilibrium that does not support the efforts required for developing highly secure software. He also argues that the software vendors do not have enough incentive to apply the available techniques to prevent software vulnerability. Until recently, low security awareness of customers has been blamed for discouraging software vendors to invest on security improvement. Since customers are not capable of predicting security loss and thus, do not appreciate the value of secure software, the vendor perceives low demand for security. As a result, developing highly secure software is a significant risk to the vendors given the small market and the high development cost.
However, low security awareness does not seem to fully explain poor software quality. Although individual consumers may be unaware of security issues, it is not the case for the firms. Especially since 9/11, security has become a major issue for business and managers have become much more aware of security concerns. In spite of increasing security awareness, the quality of software is still in question. Security researchers argue that unless software vendors are held liable for their products, they are not willing to produce secure software (Armour and Humphrey 1993, Harmon 2003, Sager and Green 2002, Schneier 2004, Varian 2000). Since the risk is entirely on the customer side, unlike other industries, such as an auto industry where the manufacturers are accountable for any defect of their products, the software vendors may not have enough incentive to develop secure software especially if the vendor dominates the entire market.

As an incentive mechanism for software security quality improvement, software liability has been intensely discussed among computer scientists, jurists, and policy makers for years. In spite of the long years of debate, the researchers have not come to a conclusion about whether imposing liability on the software vendor leads to highly secure software. In a recent article, two experts in the field take opposite viewpoints regarding software liability (Heckman 2003, Ryan 2003). Ryan (2003) argues that it is not practical for consumers to create their own security software, and thus it is reasonable to assume that manufacturers of such software should ensure the reliability of their products. On the other hand, Heckman (2003) argues that software does not have the characteristics of other products such as automobiles that would support legal liability, and thus, liability is not the appropriate tool for reducing the number and severity of software security holes.

Although the software vendors have not faced liability for security failures yet, some companies are demanding liability clauses in contracts with vendors, holding them responsible for any security breach connected to their software (Fisher 2002). Consequently, software vendors face potential threat of liability. In October 2003, Microsoft was sued in California, based on the claim that its market-dominant software is vulnerable to viruses and worms capable of triggering massive, cascading failures in global networks. The lawsuit comes in the wake of two major worms that have exploited vulnerabilities in Microsoft software: Slammer and Blaster. Due to potential liability as well as increasing customer security awareness, some leading software vendors started making efforts to develop secure software. In the now famous memo in January 2002, Bill Gates stated as follows:
“In the past, we’ve made our software and services more compelling for users by adding new features and functionality, and by making our platform richly extensible. We’ve done a terrific job at that, but all those great features won’t matter unless customers trust our software. So now, when we face a choice between adding features and resolving security issues, we need to choose security. Our products should emphasize security right out of the box, and we must constantly refine and improve that security as threats evolve.”

During February and March 2002, Microsoft stopped developing Windows features and focused on improving the security quality of Windows. For a monopolistic vendor pursuing profit such as Microsoft, this shows how serious the security problem is. This memo clearly shows that the vendor faces a new era when the development of secure software is a significant issue for business.

In this paper, we investigate whether imposing liability motivates the monopolist to improve the security quality of its software, and determine the conditions under which the liability mechanism is effective in terms of security improvement. Then we examine how the liability mechanism affects the monopolist’s market coverage. Both security quality and market coverage are important to evaluate the effectiveness of the liability mechanism since policy makers need to make sure that the monopolist has incentive to serve enough customers with providing reasonably secure software with the liability mechanism. There are two issues surrounding the liability mechanism: effectiveness and feasibility. The focus of this paper is the former, i.e., whether liability leads to higher security quality or higher social surplus. Implementing the liability mechanism is certainly not an easy problem. For example, quantifying loss is difficult. But only when the mechanism is proven to be effective, it is worth discussing the implementation issue. In spite of a long debate by computer scientists, jurists, and policy makers, there has not been an agreement on the effectiveness of the liability mechanism. From an economic perspective, we analyze the software market to evaluate the effectiveness of software liability. We model the case where the vendor focuses on security development, and identify the factors that affect the monopolist’s decision on security quality and market coverage.

We first consider the case where both the vendor and customers are fully aware of security issues.
In other words, both parties have rational expectation of the loss due to security breaches of the software. Unlike direct contractual risk-sharing that may impose partial liability on the vendor, we examine the legal and uniform liability that makes the vendor fully responsible for the entire loss. We determine the conditions that affect the monopolist’s market coverage. We find that high marginal willingness to pay for the software leads to full market coverage when the customers are responsible for the entire risk. Once liability is imposed on the monopolist, full market coverage obtains when the expected loss is bounded. Interestingly, the monopolist covering the partial market without liability may have an incentive to fully increase its market coverage when it becomes liable. The monopolist offers even higher security quality in the full market with liability than in the partial market without liability. This result implies that under certain conditions, avoiding liability is not beneficial to the software vendor in the sense that the liability mechanism may give the monopolist an opportunity to have more customers.

Overall, we find that liability does not always lead to high security quality, nor does it always lead to high social surplus. We identify the factors that affect the monopolist’s decision on security quality level. We find that no matter which party is liable for the loss in the fully covered market, security quality increases with the proportion of loss to cost and decreases with the baseline utility from the software. We also find that high marginal willingness to pay leads to high security quality in the current market without liability. This finding implies that educating the users to appreciate security may be an effective way to motivate the vendor to improve security quality. We examine the welfare implications of the liability mechanism. Our results show that security quality is underprovided in the market where customers are responsible for the entire risk while socially optimal level of security quality is offered once liability is imposed on the monopolist. We also find that the social planner offers higher security quality in the market without liability than with liability. This result implies that the social planner may need to lower the security quality to take the responsibility for the entire risk. We investigate whether imposing liability on the vendor leads to better security or higher consumer surplus in the fully covered market. Interestingly, our results indicate that imposing liability may discourage the monopolist from improving security contrary to security practitioners’ expectation, whereas consumer surplus is higher with the liability mechanism. When the marginal willingness to pay is relatively low, the liability mechanism brings higher social
surplus. Finally, we consider the case where information asymmetry exists between the software vendor and the customers. We find that the liability mechanism is effective in improving security quality as well as consumer surplus when customers are not capable of properly predicting the loss unlike the vendor. This paper contributes to the literature in that it not only gives a clear picture of liability in the software market from an economic perspective but also provides implications to managers and policy makers.

The rest of the paper is organized as follows. Section 2 briefly describes the model. In Section 3, we examine the software market without liability, and examine the conditions under which market coverage is endogenously determined. We then obtain the optimal levels of security quality of the monopolist’s software product in the partially covered market and the fully covered market. In Section 4, we analyze the market with liability. We examine the policy implications of the liability mechanism in Section 5. We analyze the impact of software liability on the levels of security quality, consumer surplus and social welfare in the fully covered market. We then investigate whether considering information asymmetry between the software vendor and the customers changes the results. Lastly, we offer our concluding remarks in Section 6.

2 Model

We analyze a software market dominated by a monopolistic software vendor, and investigate whether imposing liability gives the vendor an incentive to improve security quality of the software. We set up a model using a theoretical framework built on the models of vertical quality differentiation (Mussa and Rosen 1978, Ronnen 1991, Spence 1975). There are two types of players in the market: a monopolistic software vendor and customers. The monopoly case is relevant in the software industry. Consider the dominating position of Microsoft in the PC operating systems market, for example. Customers in our model are firms that are likely to have higher incentive to invest on security than do individuals whose security awareness level is low in reality. We first consider the case where both the vendor and customers are aware of security issues. In other words, they have rational expectations of the loss due to security breaches of the software product. \( a(1 - q) \) represents the expected loss when \( a \) is a parameter capturing the degree of the loss due to security
breaches \((a > 0)\) and \(q\) is security quality scaled between 0 and 1. Security quality measures how vulnerable the software is at the product launch. Bug-free software can be considered to be of perfect security quality \((q = 1)\). The loss function is set up in a way that increasing security quality reduces the expected loss from cyberattack in the life-span of the product, and that the loss is entirely preventable if the installed software exhibits perfect security quality. This is consistent with security experts’ view such as Fisk (2002).

Arora et al. (2003) argue that the best strategy for software vendors is to introduce the product as early as possible and then patch it later. Consequently, security quality of the software at the product launch is lower than expected. Security experts argue that the software vendors should improve the initial security quality of their products since the defects of poorly written software are exploited by the malicious hackers to attack the computer systems (Harmon 2003). Until recently, software vendors have been able to avoid liability for any defect of their products and thus, they have little incentive for security quality improvement. However, software vendors are now facing potential threat of liability as demonstrated in the Microsoft lawsuit case in October 2003. As a result, software vendors are more concerned about the security quality of their products. As noted before, in early 2002, Microsoft stopped developing Windows feature and focused on improving security quality of Windows. Our model captures the current phenomenon by analyzing the market where the vendor emphasizes security quality. \(V\) is the baseline utility that a customer gains from the software product if it is perfectly secure. For example, one may consider \(V\) as the level of functionality provided by the software. \(tVq^2\) is the cost function for security quality development where \(t\) is a parameter capturing the degree of the development cost for the software with baseline utility \(V\) and security quality \(q\) \((t > 0)\). We assume the cost to be a convex function of security quality which increases with the utility. This is realistic in the sense that it is more costly to make complicated software with more features secure.

2.1 Baseline

We examine the market where the entire risk is on the customer side, which models the current state of the software market. Our model assumes that customers have rational expectations of the loss due to the vulnerability of the software. Customers value security quality in that \(q\) reduces
expected loss. We model the case where the vendor focuses on security quality. In the market where the customers are responsible for any possible loss due to security breaches of the software, the expected utility of a customer is defined as follows:

\[ EU = \theta (V - a(1 - q)) - p. \]

Recall \( V \) denotes the baseline utility that a customer gains from the software if it is perfectly secure. \( a(1 - q) \) represents the expected loss when \( q \) is the level of security quality scaled between 0 and 1 (\( a > 0 \)). \( p \) is the price and \( \theta \) captures customer heterogeneity indicating how much utility a customer derives from the software. Losses arise out of disruption to business activities. Thus, the same attack may lead to more damage to some firms than to the others. If \( \theta \) is high, the firm cares more about security quality of the software, in that it enjoys more utility from the product, but also suffers more disutility from a successful attack. It reflects that some firms are more concerned about security issues than others. Firms in the financial industry or in health care industry are examples of high-\( \theta \) firms. Without loss of generality, we assume that \( \theta \) is uniformly distributed on \((\bar{\theta} - 1, \bar{\theta})\).

Since the entire risk is on the customers, the vendor is not responsible for any possible loss that may occur due to software vulnerabilities. The production of information good such as software involves high fixed costs but low variable costs. In other words, the cost of producing the original copy is substantial whereas the cost of making additional copies is negligible. As a result, given the context of software product, the cost does not depend on quantity. Thus, in our model, it is reasonable to assume zero variable cost. Then, the software vendor’s expected profit is

\[ E\pi = px - tVq^2 \]

where \( x \) is demand for the product and \( tVq^2 \) represents the production cost of software with security quality level \( q \) (\( t > 0 \)). This quadratic cost function implies that the cost increases as security quality level rises at a growing rate and that the cost to offer a certain level of security quality becomes higher when the software provides higher utility (\( V \uparrow \)). The assumptions are summarized as follows:

\[ q \in [0, 1] \]
\[ \theta \sim \text{Uniform}(\bar{\theta} - 1, \bar{\theta}) \]
Expected Loss \(= a(1 - q)\)

Cost \(= tVq^2\).

### 2.2 Software Liability

In this section, we model the market where liability is imposed on the vendor side. This is common in some industry. For example, car manufacturers are legally responsible not only for the product defects but also for any injury or loss that is caused by the defects. However, unlike other industry, software vendors have been able to avoid liability for any vulnerability of the product. Accordingly, security experts and jurists argue that imposing liability on the vendor may work well as an incentive mechanism for security quality improvement of the software products (Ryan 2003, Schneier 2004). We model the market where the vendor is accountable for any loss due to vulnerability of its software product. In this market, the customers do not have to take the expected loss into account when they decide whether or not to buy the software. They only care about the baseline utility and price since the expected loss does not affect total customer utility:

\[
EU = \theta V - p.
\]

In this market, where the risk is on the software vendor, the cost function for the vendor has two factors: the fixed cost for initial security quality development and the expected loss which plays a role of a variable cost. The expected profit of the vendor is then:

\[
E\pi = (p - a(1 - q))x - tVq^2.
\]

### 3 Without Liability

#### 3.1 Partial Market Coverage

We first analyze the market where the entire risk is on the customers. The game consists of three stages. At the first stage, the monopolistic vendor decides the level of security quality \(q\). At the second, the vendor sets up the price \(p\). Then the customers decide whether or not to buy the software product. Let \(\hat{\theta}\) define the marginal willingness to pay of the customer who is indifferent
between buying and not buying. Then the expected utility of the marginal customer is

\[ EU = \hat{\theta} (V - a(1 - q)) - p = 0 \]

\[ \iff \hat{\theta} = \frac{p}{V - a(1 - q)}. \]

The customers who have higher marginal willingness to pay than \( \hat{\theta} \), will buy the software product. Then the demand for the software is

\[ x = \bar{\theta} - \frac{p}{V - a(1 - q)}. \]

The expected profit for the monopolist becomes

\[ E\pi = p(\bar{\theta} - \frac{p}{V - a(1 - q)}) - tVq^2. \]

The first order condition for \( p \) is

\[ \frac{\partial E\pi}{\partial p} = \bar{\theta} - \frac{2p}{V - a(1 - q)} = 0. \]

Thus, the optimal price in the equilibrium is

\[ p^* = \frac{\bar{\theta}}{2}(V - a(1 - q)). \]

Substituting \( p^* \) in the profit function leads to

\[ E\pi = \frac{\bar{\theta}^2}{4} (V - a(1 - q)) - tVq^2. \]

The first order condition for \( q \) is

\[ \frac{\partial E\pi}{\partial q} = \frac{a\bar{\theta}^2}{4} - 2tVq = 0. \]

Let 1 denote the case without liability and \( P \) in the subscript represent partial market coverage. Solving the FOC yields the optimal security quality as follows:

\[ q_{1P}^* = \frac{a\bar{\theta}^2}{8tV}. \]

The optimal level of price is then

\[ p_{1P}^* = \frac{\bar{\theta}}{2}(V - a(1 - \frac{a\bar{\theta}^2}{8tV})). \]
3.2 Full Market Coverage

In this section, we examine the conditions under which the market is fully covered by the monopolist and then analyze the monopolist’s decision on security quality in the fully covered market. In the context of software, this is a relevant case since full market coverage is quite often achieved in reality. When \( x \geq 1 \) in the partial market, full market coverage obtains from the outset. We first analyze the market without liability. In Section 3.1, we obtained the demand in the partially covered market as follows:

\[
x_{1P} = \bar{\theta} - \frac{p}{V - a(1-q)}.
\]

The optimal level of price is then

\[
p^*_1 = \frac{\bar{\theta}}{2} (V - a(1-q)).
\]

Substituting \( p \) in the demand with the equilibrium price yields

\[
x^*_1 = \frac{\bar{\theta}}{2}.
\]

Thus the full market condition for the market without liability is

\[
x^*_1 \geq 1
\]

\[\iff \bar{\theta} \geq 2.
\]

The result leads to the following proposition.

**Proposition 1** *In the market without liability, the monopolistic software vendor covers the full market when the customers have high marginal willingness to pay (\( \bar{\theta} \geq 2 \)). When the customers have low marginal willingness to pay (\( 1 \leq \bar{\theta} < 2 \)), some customers are left out.*

When the customers are responsible for the entire risk and the marginal willingness to pay is high, the monopolist has an incentive to cover the full market. Otherwise, the monopolist covers only the partial market. This is quite intuitive in the sense that full market coverage obtains when the customers value the product more. Proposition 1 shows that the marginal willingness to pay for the software is the only factor that decides the market coverage when the software vendors can avoid liability.
Let \( \hat{\theta} \) define the marginal willingness to pay of the customer who is indifferent between buying and not buying. Then
\[
\hat{\theta} = \frac{p}{V - a(1 - q)}.
\]
Assume that \( \bar{\theta} \geq 2 \). Then
\[
x = 1 \text{ for all } p, V, q, a \text{ such that } \hat{\theta} \leq \bar{\theta} - 1.
\]
Then the monopolist charges the highest possible price, \( \bar{p} \):
\[
\hat{\theta} \leq \bar{\theta} - 1 \\
\Leftrightarrow \frac{p}{V - a(1 - q)} \leq \bar{\theta} - 1 \\
\Leftrightarrow p \leq (\bar{\theta} - 1)(V - a(1 - q)) \\
\Leftrightarrow \bar{p} = (\bar{\theta} - 1)(V - a(1 - q)).
\]
Under full market coverage, the monopolist’s expected profit is
\[
E\pi = \bar{p} - tq^2 \\
= (\bar{\theta} - 1)(V - a(1 - q)) - tq^2.
\]
The first order condition for \( q \) is
\[
\frac{\partial E\pi}{\partial q} = a(\bar{\theta} - 1) - 2tVq = 0.
\]
Recall that 1 denotes the case without liability. Let \( F \) in the subscript stand for full market coverage.
Solving the first order condition yields the optimal security quality:
\[
q^*_F = \frac{a(\bar{\theta} - 1)}{2tV}.
\]
The optimal level of price is then
\[
p^*_F = (\bar{\theta} - 1)(V - a(1 - \frac{a(\bar{\theta} - 1)}{2tV})).
\]
The monopolist’s expected profit in the fully covered market without liability is
\[
E\pi^*_F = (\bar{\theta} - 1)(V - a(1 - \frac{a(\bar{\theta} - 1)}{2tV})) - t\frac{a^2(\bar{\theta} - 1)^2}{4tV} \\
= (\bar{\theta} - 1)(V - a) + \frac{a^2(\bar{\theta} - 1)^2}{4tV}.
\]
4 With Liability

4.1 Partial Market Coverage

We analyze the market where the vendor is liable for the security breaches of its software product, assuming all losses are verifiable. Under the liability mechanism, the vendor is supposed to cover the entire loss incurred due to the attacker’s exploitation of the software vulnerability. Since the risk is not on the customers but on the vendor, the utility function of customers is not affected by the expected loss. The customer’s utility function is

\[ EU = \theta V - p. \]

Although the variable cost of production is assumed to be zero, the liability mechanism makes the expected loss be the variable cost. Thus, the vendor’s expected profit function is

\[ E\pi = (p - a(1 - q))x - tVq^2. \]

Then the willingness to pay of the marginal customer is

\[ EU = \hat{\theta}V - p = 0 \]
\[ \iff \hat{\theta} = \frac{p}{V}. \]

The customers who have higher marginal willingness to pay than \( \hat{\theta} \) will buy the software product whereas others will not. Thus, the demand is

\[ x = \bar{\theta} - \frac{p}{V}. \]

The expected profit for the monopolist who is liable for the loss is

\[ E\pi = (p - a(1 - q))(\bar{\theta} - \frac{p}{V}) - tVq^2. \]

The first order condition for the price is

\[ \frac{\partial E\pi}{\partial p} = \hat{\theta} - \frac{2p}{V} + \frac{a(1 - q)}{V} = 0. \]

The optimal price level is

\[ p^* = \frac{V\hat{\theta} + a(1 - q)}{2}. \]
Replacing price in the expected profit with $p^*$ yields

$$E \pi = \frac{1}{4V}(V \bar{\theta} - a(1 - q))^2 - tV q^2.$$  

Then the first order condition for $q$ is

$$\frac{\partial E \pi}{\partial q} = \frac{a}{2V}(V \bar{\theta} - a(1 - q)) - 2tV q = 0$$
$$\Leftrightarrow q(a^2 - 4tV^2) = a^2 - aV \bar{\theta}.$$

Let 2 denote the case with liability. The optimal security quality is

$$q^{\ast}_2 = \frac{a^2 - aV \bar{\theta}}{a^2 - 4tV^2}.$$  

**Proposition 2** When the entire risk is on the customer side in the partially covered market, security quality ($q^*$) increases with the proportion of loss to cost ($a/t$) and marginal willingness to pay ($\bar{\theta}$) and decreases with utility ($V$). However, when liability is imposed on the vendor, security quality does not always increase with marginal willingness to pay. Only when the expected loss is relatively low ($a < 2\sqrt{tV}$), higher marginal willingness to pay leads to higher security quality:

$$q^{\ast}_2 \uparrow \text{ as } \bar{\theta} \uparrow \text{ when } 0 < a < 2\sqrt{tV}$$
$$q^{\ast}_2 \downarrow \text{ as } \bar{\theta} \uparrow \text{ when } a > 2\sqrt{tV}.$$

Without a liability mechanism, the monopolist improves the security quality with the proportion of loss to cost ($a/t$), which can be interpreted as the efficiency of the security development. We find that higher marginal willingness to pay leads to higher security quality, implying that educating customers to appreciate security may be an effective way to motivate the software vendor to improve the security quality of its software. Once liability is imposed on the vendor, the security quality does not always increase with the marginal willingness to pay since the possible loss due to the security breaches does not directly affect customer’s utility. Our results show that the security quality increases with the marginal willingness to pay when the expected loss is relatively small. Interestingly, the software vendor reduces the security quality as the customers increase their valuation of the software when the expected loss is large. The results imply that the vendor may reduce the demand to minimize the total expected loss by lowering the security quality when it perceives high expected loss.
4.2 Full Market Coverage

In this section, we analyze the market with liability and examine the conditions under which the market is fully covered by the monopolist. We model the market where the vendor is responsible for the entire loss due to security breaches which are exploited by malicious agents. The full market coverage condition is \( x \geq 1 \). The demand for the software in the partially covered market is

\[
x_{2P} = \bar{\theta} - \frac{p}{V}.
\]

In Section 4.1, we derived the equilibrium price as follows:

\[
p_{2P}^* = \frac{V \bar{\theta} + a(1 - q)}{2}
\]

Note that the optimal quality is

\[
q_{2P}^* = \frac{a^2 - aV \bar{\theta}}{a^2 - 4tV^2}.
\]

We can compute the demand by substituting price and quality with optimal levels.

\[
x_{2P} = \frac{\bar{\theta}}{2} - \frac{a(1 - q_{2P}^*)}{2V} = \frac{1}{2V} \left( \bar{\theta}V - \frac{a^2V \bar{\theta} - 4atV^2}{a^2 - 4tV^2} \right).
\]

The full market condition for the market with liability is then

\[
x_{2P} = \frac{1}{2V} \left( \bar{\theta}V - \frac{a^2V \bar{\theta} - 4atV^2}{a^2 - 4tV^2} \right) \geq 1
\]

\[
\Leftrightarrow \frac{(\bar{\theta}V - 2V)(a^2 - 4tV^2) - a^2V \bar{\theta} + 4atV^2}{a^2 - 4tV^2} \geq 0.
\]

**Proposition 3** In the market with liability, the monopolistic software vendor has an incentive to cover the full market for any level of marginal willingness to pay of the customers. Full market coverage obtains only when the expected loss is bounded either by \( 2\sqrt{tV} \) or \( a_+ \) where

\[
a_+ = tV \left( 1 + \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right).
\]
Recall that in the market without liability, high marginal willingness to pay gives the monopolist an incentive to cover the full market. Proposition 3 implies that the monopolist is willing to cover the full market given any level of the marginal willingness to pay in the market with liability. Depending on the conditions, full market coverage obtains even for the customers with low marginal willingness to pay. This is interesting in the sense that the monopolist who avoids liability and serves partial market may want to increase the market coverage even when liable. Thus, the liability mechanism may give an opportunity for the monopolist to serve more customers who may be better off since they do not have to worry about the possible loss any more. When the marginal willingness to pay is high ($\bar{\theta} > 2 + \frac{t}{2}$) in the market with liability, the monopolist covers the full market if the expected loss is relatively small, bounded by $2\sqrt{tV}$. If the monopolist perceives high loss, it only serves the partial market. In case the marginal willingness to pay is low ($1 < \bar{\theta} \leq 2 + \frac{t}{2}$), full market coverage obtains only when the expected loss is bounded by either $2\sqrt{tV}$ or $a_+$. Interestingly, no matter what level of the baseline utility is given, the monopolist covers the full market only when the expected loss is bounded. Since the vendor is responsible for the entire loss, it has an incentive to cover the full market only when the maximum possible loss is bounded. In the market without liability, the monopolist covers the full market when customer valuation is high no matter how high the maximum possible loss is. However, once liability is imposed on the vendor, the vendor never serves the full market when it faces catastrophic loss.

Suppose that the full market condition holds. Let $\hat{\theta}$ be the marginal willingness to pay of the customer who is indifferent between buying and not buying:

$$\hat{\theta} = \frac{p}{V}.$$  

Consider the case where $\bar{\theta}$ and $a$ meet the full market condition. Then

$$x = 1$$

for all $p, V, q, a$ such that $\hat{\theta} \leq \bar{\theta} - 1$.

Then the monopolist charges the highest possible price, $\bar{p}$:

$$\bar{p} = (\bar{\theta} - 1)V.$$  

Under full market coverage, the monopolist maximizes the expected profit:

$$E\pi = \bar{p} - a(1 - q) - tVq^2$$
\[
= (\bar{\theta} - 1)V - a(1 - q) - tVq^2.
\]

The first order condition for \( q \) is
\[
\frac{\partial E\pi}{\partial q} = a - 2tVq = 0.
\]

Recall that 2 denotes the case with liability. Then solving the first order condition yields the optimal security quality:
\[
q^*_2 = \frac{a}{2tV}.
\]

Note that
\[
p^*_2 = (\bar{\theta} - 1)V
\]

The monopolist’s expected profit in the fully covered market with liability is
\[
E\pi^*_{2F} = (\bar{\theta} - 1)V - a(1 - \frac{a}{2tV}) - \frac{a^2}{4tV}
\]
\[
= (\bar{\theta} - 1)V - a + \frac{a^2}{4tV}.
\]

**Proposition 4** No matter which party is liable for the loss in the fully covered market, security quality \((q^*)\) increases with the proportion of loss to cost \((a/t)\) and decreases with baseline utility \((V)\).

When the risk is entirely on the customer, security quality increases with the marginal willingness to pay \((\bar{\theta})\) which does not affect security quality under liability mechanism.

The findings are consistent with the security experts’ argument that it is harder to make the software with more features secure since breaches are more likely to be found in such software. Bruce Schneier, founder and chief technology officer of Internet security firm Counterpane said, “Complexity is the enemy of security. As systems get more complex, they get less secure,” in an interview with PCWorld.com (Zetter 2001). Proposition 4 supports the argument that software vendors often sacrifice security quality to provide high functionality. We also find that when loss is relatively large compared to cost \((a/t \uparrow)\), meaning that investment on security development is efficient, the monopolist offers high security quality no matter what mechanism is adopted. In other words, the vendor is willing to produce software with higher security quality when the same amount of investment is expected to prevent larger loss. Our results show the interplay of the marginal willingness to pay for the software and the optimal level of security quality. When there
is no vendor liability, customers must value the software more to get higher security. However, once liability is imposed on the vendor, marginal willingness to pay does not affect security quality. This is because the customers do not appreciate security quality when the entire risk is on the vendor side.

**Proposition 5** When full market coverage obtains with or without a liability mechanism, imposing liability on the monopolist reduces security quality \(q_{2F}^* < q_{1F}^*\) and increases price \(p_{2F}^* > p_{1F}^*\). The liability mechanism allows higher profit to the monopolist when the marginal willingness to pay is low while it leads to lower profit when the marginal willingness to pay is high:

\[
E\pi_{2F}^* > E\pi_{1F}^* \quad \text{if} \quad 2 < \bar{\theta} < \frac{4tV}{a},
\]
\[
E\pi_{2F}^* < E\pi_{1F}^* \quad \text{if} \quad \bar{\theta} > \frac{4tV}{a}.
\]

This is interesting in the sense that unlike the security experts’ belief that the liability mechanism gives the software vendors an incentive to improve security quality (Armour and Humphrey 1993), imposing liability may discourage the vendor to develop security in the fully covered market. In the current market where customers are responsible for the entire risk, the expected loss directly affects customer valuation as well as vendor cost. When liability is imposed on the vendor, customers do not have to care about the expected loss, and they do not appreciate the vendor’s effort for security development. Thus, the expected loss only affects the vendor’s development cost. Consequently, the vendor has a higher incentive to develop security to maximize its profit when customers are liable. When liability is imposed on the monopolist, it reduces the security quality but charges higher price for its software. The monopolist gets higher profit in the market with liability when the marginal willingness to pay of the customers is relatively low.

**Proposition 6** The monopolist covering partial market without liability may have an incentive to cover the full market with liability by increasing quality \(q_{2F}^* > q_{1P}^*\).

Recall that the monopolist who covers the partial market without liability may have an incentive to cover the full market once liability is imposed. Proposition 6 shows that the monopolist offers
higher security quality in the full market with liability than in the partial market without liability. This result implies that the monopolist may want to minimize the expected loss by improving security quality to increase market coverage when it is liable for the loss. Since the loss directly affects the monopolist’s profit in the market with liability, the monopolist may have an incentive to offer more secure software than one in the current market without liability where the loss is not a concern of the vendor.

5 Policy Implication

5.1 Welfare Analysis

We examine the social planner’s decision on the security quality and investigate whether the monopolist offers socially optimal level of quality. First, we examine the case where no liability is imposed on the software vendor. Unlike the monopolist who maximizes the profit, the social planner maximizes the social welfare by pricing at the marginal cost, which is zero. Recall that 1 denotes the case without liability. The social surplus can be written as the difference between the gross consumer surplus and the production cost:

\[ S_1 = \int_{\theta-1}^{\bar{\theta}} \theta(V - a(1 - q))d\theta - tVq^2 \]

\[ = \frac{1}{2}(2\bar{\theta} - 1)(V - a(1 - q)) - tVq^2. \]

The first order condition for \( q \) is as follows:

\[ \frac{\partial S_1}{\partial q} = a\frac{2(2\bar{\theta} - 1)}{2tV}. \]

Let \( SP \) represent the social planner’s solution. The optimal security quality of the social planner is

\[ q_{1SP}^* = \frac{a(\bar{\theta} - \frac{1}{2})}{2tV}. \]

We examine the market where liability is imposed on the software vendor. We consider the social planner who covers the full market, and maximizes the social surplus by pricing at the marginal cost which is the expected loss in this case. Recall that 2 denotes the case with liability. Then the social planner in the fully covered market with a liability mechanism maximizes the following social

18
surplus:

\[ S_2 = \int_{\tilde{\theta} - 1}^{\tilde{\theta}} \theta V d\theta - a(1 - q) - tVq^2 \]
\[ = \frac{1}{2}(2\tilde{\theta} - 1)V - a(1 - q) - tVq^2. \]

The first order condition for \( q \) is then:

\[ \frac{\partial S_2}{\partial q} = a - 2tVq = 0. \]

Thus, the optimal security quality of the social planner is

\[ q^*_{2SP} = \frac{a}{2tV}. \]

**Proposition 7** When the market is fully covered, the social planner offers higher security quality in the market without liability than with liability. Without liability, security quality is underprovided by the monopolist compared to the social optimum. However, once liability is imposed, the monopolist offers socially optimal level of security quality.

We find that the social planner offers higher security quality in the market without liability than with liability. This implies that the social planner may need to lower the security quality to take the responsibility for the entire risk. Proposition 7 provides an evidence of underprovided security quality of software under monopoly, as has been observed in the current market. It shows that imposing liability on the monopolist may give an incentive to offer socially optimal level of security quality. This result has a policy implication in that liability should be imposed on the vendor to achieve socially optimal level of security when the market is fully covered by the monopolist.

Recall that when the full market condition holds for both mechanisms, imposing liability may lead to worse security quality. We investigate whether a liability mechanism is beneficial to customers and to the entire society. Under the mechanism in the current market, customers are responsible for the entire risk which is not a concern for the vendor. However, once liability is imposed on the vendor, the customers do not take the loss into account. The comparison of consumer surplus is not trivial in that although liability mechanisms may make customers free from loss, it leads to higher price than the current market.
The consumer surplus in the market without liability is

\[ CS_{1F} = \int_{\bar{\theta} - 1}^{\bar{\theta}} \theta(V - a(1 - q))d\theta - \bar{p} \]

\[ = \frac{1}{2}(2\bar{\theta} - 1)(V - a(1 - q)) - (\bar{\theta} - 1)(V - a(1 - q)) \]

\[ = \frac{1}{2}(V - a(1 - q)). \]

Substituting \( q \) with the equilibrium quality leads to the following consumer surplus:

\[ CS_{1F} = \begin{cases} 
\frac{1}{2} \left(V - a + \frac{a^2(\bar{\theta} - 1)}{2V}\right) & \text{if } \frac{a(\bar{\theta} - 1)}{2V} < 1 \\
\frac{1}{2} V & \text{if } \frac{a(\bar{\theta} - 1)}{2V} \geq 1.
\end{cases} \]

Consider the case where liability is imposed on the monopolist. Then the consumer surplus can be written as

\[ CS_{2F} = \int_{\bar{\theta} - 1}^{\bar{\theta}} \theta V d\theta - \bar{p} \]

\[ = \frac{1}{2} (2\bar{\theta} - 1) - (\bar{\theta} - 1)V \]

\[ = \frac{1}{2} V. \]

The following proposition compares consumer surplus in both mechanisms.

**Proposition 8** When full market coverage obtains with or without liability, imposing liability on the monopolist may increase consumer surplus. When the marginal willingness to pay is relatively low \((2 < \bar{\theta} < \frac{4V}{a})\), the liability mechanism leads to higher social surplus as well.

Schneier (2004) argues that the customers will pay for security once liability is imposed on the software vendor, which means security is not free even in the market where the vendor is liable. Since the vendor passes some portion of security development cost onto the customers, it is not trivial whether consumer surplus becomes higher because of customers being free from the loss or lower due to the increased price. In the fully covered market, we find that the monopolist reduces security quality when liable. Unlike security quality, consumer surplus becomes higher when liability is imposed on the vendor. Although the customers will pay for security, the results indicate that the customers are better off under the liability mechanism. When the marginal willingness to pay is low, the liability mechanism brings higher social surplus as well.
5.2 Information Asymmetry

Varian (2000) argues that liability should be assigned to the party that can do the best job of managing risk. Other security experts who champion the liability mechanism argue that liability should be imposed on the software vendor since the vendor is the only party to have an ability to understand the vulnerabilities, to predict the possible damage due to the defects, and to know how to fix them (Ryan 2003). In other words, customers are not perfectly aware of security issues, and thus they are not able to predict possible loss. As a result, they tend to underestimate the possible damages due to software defects. In this section, we investigate whether this information asymmetry between the vendor and customers changes our results. We model this asymmetry by incorporating a parameter $r$, which denotes security awareness of the customers. Then the expected loss perceived by the customers is

$$ar(1-q) \quad \text{where} \quad r < 1.$$ 

So far, we investigated the case where $r = 1$. Note that the parameter $r$ does not affect the optimal quality in the market with liability since the entire loss is borne by the vendor, and thus it is not in the customer’s utility function. After going through some algebra, we can obtain the optimal quality in the market without liability. The optimal security qualities are as follows:

$$q_{1F}^* = \frac{ar(\theta - 1)}{2tV},$$

$$q_{2F}^* = \frac{a}{2tV}.$$ 

**Proposition 9** When full market coverage obtains with or without liability and the customers underestimate the expected loss ($r \leq \frac{1}{\bar{\theta} - 1}$), imposing liability on the monopolist may increase security quality as well as consumer surplus.

Proposition 9 supports security experts’ arguments that imposing liability on the software vendor gives an incentive for security quality improvement especially when the customers are not capable of understanding the risks and predicting the loss due to software vulnerabilities (Ryan 2003, Schneier 2004, Varian 2000). We find that unlike the case where customers are fully aware of security issues, liability mechanism is effective in terms of quality improvement as well as consumer surplus increase.
No matter how much customers are aware of security, a liability mechanism makes the customers better off by increasing consumer surplus. Moreover, particularly when the customers underestimate security concerns, the liability mechanism is what policy makers need to consider to have software products more secure.

6 Concluding Remarks

Low quality of software has been blamed for poor security of our computer networks in the sense that the defects of poorly written software are exploited by malicious hackers to attack the computer systems. As an incentive mechanism for software quality improvement, software liability has been intensely discussed among computer scientists, jurists and policy makers (Armour and Humphrey 1993, Harmon 2003, Heckman 2003, Ryan 2003, Schneier 2004, Varian 2000). Unfortunately, not much research has been done on this issue from an economic perspective. In this paper, we analyze the software market dominated by a monopolistic vendor, and investigate whether imposing liability motivates the monopolist to improve the security quality of the software, and makes the customers and the entire society better off. We model the case where the vendor focuses on security quality improvement and both the vendor and customers are fully aware of security issues. In other words, both parties have rational expectations of the loss caused by security breaches of the software. Unlike direct contractual risk-sharing that may impose partial liability on the vendor, we examine the legal and uniform liability that makes the vendor fully responsible for the entire loss incurred by security breaches of its software. We identify the factors that affect the monopolist’s decision on the security quality.

We determine the conditions under which the monopolist has an incentive to cover the full market. We find that high marginal willingness to pay leads to full market coverage when the customers are responsible for the entire risk. Once liability is imposed on the monopolist, full market coverage obtains when the expected loss is bounded. Our welfare analysis shows that the social planner offers higher security quality in the market without liability than with liability and that security quality is underprovided by the monopolist in the market where customers are responsible for the entire risk while socially optimal level of security quality is offered once liability is imposed on the monopolist.
Interestingly, our results indicate that imposing liability may discourage the monopolist improving security contrary to security practitioners’ expectation, whereas consumer surplus is higher with a liability mechanism. When the marginal willingness to pay is relatively low, the liability mechanism leads to higher social surplus. Finally, we consider the case where information asymmetry exists between the software vendor and the customers. We find that the liability mechanism leads to higher security quality as well as higher consumer surplus in the presence of information asymmetry between the software vendor and the customers.

Several limitations of our study deserve further discussion. First, our model does not consider patch management which can be an alternative way of achieving security. Setting up a multi-period model and analyzing how the endogenous patching quality can affect the initial security quality will be an interesting question for future research. Second, the implicit assumption in our model is that the expected loss can be estimated. However, this is not an easy problem. Quantifying loss itself is a hard problem and many researchers have been working on this issue, but there is no standardized method as yet. Security experts argue that the prerequisite for the software liability mechanism is measuring the loss and estimating the likelihood. Despite these limitations, our paper contributes to the literature in the following ways. First, our paper analyzes the problem of software liability from an economic perspective where not much economic research has been done on this issue although it has been intensely argued by computer scientist, jurists, and policy makers. Second, we provide a clear picture of the software market, and give a guideline to managers and policy makers who try to make software secure. Our results help policy makers decide which mechanism they need to choose under certain conditions. Third, our paper explains the huge security investment of the monopolistic software vendors by identifying the factors that affect the vendors’ decision on security quality such as baseline utility of the software product, security awareness of the customers, and the potential threat to be liable for the loss caused by the software vulnerabilities.
Appendix

Proof of Proposition 2

Let \( k = a/t \). Then

\[
\begin{align*}
\frac{\partial q_{1P}^*}{\partial k} &= \frac{\bar{\theta}^2}{8V} > 0 \\
\frac{\partial q_{1P}^*}{\partial \bar{\theta}} &= \frac{a\bar{\theta}}{4tV} > 0 \\
\frac{\partial q_{1P}^*}{\partial V} &= -\frac{a\bar{\theta}^2}{8tV^2} < 0.
\end{align*}
\]

Note that

\[
\begin{align*}
\frac{\partial q_{2P}^*}{\partial \bar{\theta}} &= \frac{aV}{4tV^2 - a^2} > 0 \text{ when } a^2 < 4tV^2 \\
\frac{\partial q_{2P}^*}{\partial \bar{\theta}} &= \frac{aV}{4tV^2 - a^2} < 0 \text{ when } a^2 > 4tV^2.
\end{align*}
\]

Since \( V > 0 \),

\[
\begin{align*}
a^2 < 4tV^2 &\iff 0 < a < 2\sqrt{t}V \\
a^2 > 4tV^2 &\iff a > 2\sqrt{t}V.
\end{align*}
\]

QED.

Proof of Proposition 3

Note that full market coverage obtains when

\[
\frac{(\bar{\theta}V - 2V)(a^2 - 4tV^2) - a^2V\bar{\theta} + 4atV^2}{a^2 - 4tV^2} \geq 0.
\]

Case 1: High marginal willingness to pay (\( \bar{\theta} > 2 + \frac{t}{2} \))

1-1: Low loss (\( 0 < a < 2\sqrt{t} \))

Note that \( a^2 - 4tV^2 < 0 \). The full market condition can be written as

\[
(\bar{\theta}V - 2V)(a^2 - 4tV^2) - a^2V\bar{\theta} + 4atV^2 \leq 0 \\
\iff a^2 - 2atV + 2tV^2(\bar{\theta} - 2) \geq 0.
\]

Consider the following equation:

\[
a^2 - 2atV + 2tV^2(\bar{\theta} - 2) = 0.
\]
The solutions are

\[ a_- = tV \left( 1 - \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right) \quad \text{and} \quad a_+ = tV \left( 1 + \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right). \]

Note that

\[ \bar{\theta} > 2 + \frac{t}{2} \iff 1 - \frac{2}{t} \bar{\theta} + \frac{4}{t} < 0. \]

Thus, \( a_- \) and \( a_+ \) do not exist, meaning that the full market condition always holds:

\[ a^2 - 2atV + 2tV^2(\bar{\theta} - 2) > 0. \]

1-2: High loss \((a > 2\sqrt{t})\)

Note that \( a^2 - 4tV^2 > 0 \). The full market condition is \( a^2 - 2atV + 2tV^2(\bar{\theta} - 2) \leq 0 \). Since \( a_- \) and \( a_+ \) do not exist, \( a^2 - 2atV + 2tV^2(\bar{\theta} - 2) > 0 \). Thus, the monopolist covers the partial market in this case.

Case 2: Medium marginal willingness to pay \((2 < \bar{\theta} < 2 + \frac{t}{2})\)

Note that \( 2 < \bar{\theta} < 2 + \frac{t}{2} \). Thus,

\[
\begin{align*}
0 < &\sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} < 1 \\
\iff & 0 < tV \left( 1 - \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right) < tV \\
\iff & 0 < a_- < tV
\end{align*}
\]

2-1: Low loss \((0 < a < 2\sqrt{t})\)

Then \( a^2 - 4tV^2 < 0 \). The full market condition is \( a^2 - 2atV + 2tV^2(\bar{\theta} - 2) \geq 0 \). The monopolist covers the full market when

\[ a \leq a_- \quad \text{or} \quad a \geq a_+. \]

2-2: High loss \((a > 2\sqrt{t})\)

Then \( a^2 - 4tV^2 > 0 \). The condition for full market coverage is

\[ a_- \leq a \leq a_+. \]

Case 3: Low marginal willingness to pay \((1 < \bar{\theta} < 2)\)

Since \( 1 < \bar{\theta} < 2 \),

\[ 1 < \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} < \sqrt{1 + \frac{2}{t}}. \]
Then we have
\[ tV \left( 1 - \sqrt{1 - \frac{2}{t} - \frac{4}{t} \bar{\theta}} \right) < 0 \]
\[ \iff a_0 < 0. \]

3-1: Low loss \((0 < a < 2\sqrt{t})\)
Then \(a^2 - 4tV^2 < 0\). The full market condition is
\[ a^2 - 2atV + 2tV^2(\bar{\theta} - 2) \geq 0. \]
The monopolist covers the full market when \(a \geq a_+\).

3-2: High loss \((a > 2\sqrt{t})\)
Then \(a^2 - 4tV^2 > 0\). The condition for full market coverage is \(0 \leq a \leq a_+\).
The results can be summarized in Table 1.

<table>
<thead>
<tr>
<th>(a)</th>
<th>(\bar{\theta} &gt; 2 + \frac{t}{2})</th>
<th>(2 &lt; \bar{\theta} &lt; 2 + \frac{t}{2})</th>
<th>(1 &lt; \bar{\theta} &lt; 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0 &lt; 2\sqrt{t}V &lt; a_0)</td>
<td>Full Market if (a \in (0, 2\sqrt{t}V)) or (a \in (a_-, a_+)^+)</td>
<td>Full Market if (a \in (0, 2\sqrt{t}V)) or (a \in (a_-, a_+)^+)</td>
<td>N/A</td>
</tr>
<tr>
<td>(a_0 &lt; 2\sqrt{t}V &lt; a_+)</td>
<td>Full Market if (a \in (0, 2\sqrt{t}V)) or (a \in (2\sqrt{t}V, a_+)^+)</td>
<td>Full Market if (a \in (0, a_-)) or (a \in (2\sqrt{t}V, a_+)^+)</td>
<td>Full Market if (a \in (2\sqrt{t}V, a_+)^+)</td>
</tr>
<tr>
<td>(a_+ &lt; 2\sqrt{t})</td>
<td>Full Market if (a \in (0, 2\sqrt{t}V)) or (a \in (a_+, 2\sqrt{t}V))</td>
<td>Full Market if (a \in (0, a_-)) or (a \in (a_+, 2\sqrt{t}V))</td>
<td>Full Market if ((a_+, 2\sqrt{t}V))</td>
</tr>
</tbody>
</table>

Therefore, the monopolist has an incentive to cover the full market for any level of marginal willingness to pay. The full market obtains only when the expected loss is bounded either by \(2\sqrt{t}V\) or \(a_+\). That is, when the expected loss is unbounded, the monopolist always cover the partial market.

QED.

Proof of Proposition 4

Note that
\[ q_{1F}^* = \frac{a(\bar{\theta} - 1)}{2tV} \quad \text{and} \quad q_{2F}^* = \frac{a}{2tV}. \]
Let \( k = a/t \). Then
\[
\frac{\partial q^*_1}{\partial k} = \frac{(\bar{\theta} - 1)}{2V} > 0
\]
\[
\frac{\partial q^*_1}{\partial V} = -\frac{a(\bar{\theta} - 1)}{2tV^2} < 0
\]
\[
\frac{\partial q^*_1}{\partial \theta} = \frac{a}{2tV} > 0
\]
and
\[
\frac{\partial q^*_2}{\partial k} = \frac{1}{2V} > 0
\]
\[
\frac{\partial q^*_2}{\partial V} = -\frac{a}{2tV^2} < 0.
\]

QED.

**Proof of Proposition 5**

Note that
\[
q^*_1 F = \frac{a(\bar{\theta} - 1)}{2tV} \quad \text{and} \quad q^*_2 F = \frac{a}{2tV}.
\]

Thus,
\[
q^*_1 F = 1 \quad \text{if} \quad \frac{a(\bar{\theta} - 1)}{2tV} \geq 1
\]
\[
q^*_2 F = 1 \quad \text{if} \quad \frac{a}{2tV} \geq 1.
\]

No matter which mechanism is adopted in the market, the monopolist covers the full market when
\[
\bar{\theta} \geq 2, \quad V > \frac{a}{2\sqrt{t}} \quad \text{and} \quad a \leq a_- \quad \text{or} \quad a \geq a_+.
\]

or
\[
2 \leq \bar{\theta} \leq 2 + \frac{t}{2}, \quad V < \frac{a}{2\sqrt{t}} \quad \text{and} \quad a_- \leq a \leq a_+.
\]

where
\[
a_- = tV \left( 1 - \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right) \quad \text{and} \quad a_+ = tV \left( 1 + \sqrt{1 - \frac{2}{t} \bar{\theta} + \frac{4}{t}} \right).
\]

Since \( \bar{\theta} \geq 2, \)
\[
1 < \frac{a}{2tV} \quad \Rightarrow \quad q^*_1 F = q^*_2 F = 1
\]
\[
\frac{a}{2tV} \leq 1 \leq \frac{a(\bar{\theta} - 1)}{2tV} \quad \Rightarrow \quad q^*_1 F = 1 \geq \frac{a}{2tV} = q^*_2 F
\]
\[
\frac{a(\bar{\theta} - 1)}{2tV} < 1 \quad \Rightarrow \quad q^*_1 F = \frac{a(\bar{\theta} - 1)}{2tV} \geq \frac{a}{2tV} = q^*_2 F.
\]
Therefore, $q_{1F}^* \geq q_{2F}^*$. 

Note that $p_{1F}^* = (\bar{\theta} - 1)(V - a(1 - q_{1F}^*))$ and $p_{2F}^* = (\bar{\theta} - 1)V$. 

Then,

$$p_{2F}^* - p_{1F}^* = a(1 - q_{1F}^*).$$

Since $q_{1F}^* \leq 1$,

$$a(1 - q_{1F}^*) \geq 0.$$

Therefore, $p_{2F}^* \geq p_{1F}^*$. 

Recall that

$$E\pi_{1F}^* = (\bar{\theta} - 1)(V - a) + \frac{a^2(\bar{\theta} - 1)^2}{4tV}$$

$$E\pi_{2F}^* = (\bar{\theta} - 1)V - a + \frac{a^2}{4tV}.$$

Thus,

$$E\pi_{2F}^* - E\pi_{1F}^* = a(\bar{\theta} - 2)(\frac{a}{4tV} \bar{\theta} - 1).$$

Therefore,

$$E\pi_{2F}^* > E\pi_{1F}^* \quad \text{if} \quad 2 < \bar{\theta} < \frac{4tV}{a}$$

$$E\pi_{2F}^* < E\pi_{1F}^* \quad \text{if} \quad \bar{\theta} > \frac{4tV}{a}.$$

QED.

**Proof of Proposition 6**

The monopolist covers the partial market without liability when $1 < \bar{\theta} < 2$. With liability, full market obtains for any $1 < \bar{\theta} < 2$, when the following conditions hold:

$$V > \frac{a}{2\sqrt{t}} \quad \text{and} \quad a \leq a_- \text{ or } a \geq a_+$$

or

$$V < \frac{a}{2\sqrt{t}} \quad \text{and} \quad a_- \leq a \leq a_+.$$

Note that

$$q_{1P}^* = \frac{a\bar{\theta}^2}{8tV} \quad \text{and} \quad q_{2F}^* = \frac{a}{2tV}.$$
Since $1 < \bar{\theta} < 2$,
\[ q_1^* = \frac{a\bar{\theta}^2}{8tV} < \frac{a}{2tV} = q_2^* . \]
\textbf{QED.}

\textbf{Proof of Proposition 7}

Note that $\bar{\theta} > 2$. Thus,
\[ q_{1SP}^* = \frac{a(\bar{\theta} - \frac{1}{2})}{2tV} > q_{2SP}^* = \frac{a}{2tV} . \]
Then the following completes the proof.
\[ q_{1F}^* = \frac{a(\bar{\theta} - 1)}{2tV} < \frac{a(\bar{\theta} - \frac{1}{2})}{2tV} = q_{1SP}^* \]
\[ q_{2F}^* = \frac{a}{2tV} = q_{2SP}^* . \]
\textbf{QED.}

\textbf{Proof of Proposition 8}

Note that
\[ \frac{a(\bar{\theta} - 1)}{2tV} \geq 1 \Rightarrow CS_{1F} = \frac{1}{2} V = CS_{2F} \]
\[ \frac{a(\bar{\theta} - 1)}{2tV} < 1 \Rightarrow CS_{1F} - CS_{2F} = \frac{a}{2} \left( \frac{a(\bar{\theta} - 1)}{2tV} - 1 \right) < 0 \]
\[ \Leftrightarrow CS_{1F} < CS_{2F} . \]
Thus,
\[ CS_{1F} \leq CS_{2F} . \]

In Proposition 5, we show that
\[ E\pi_{1F} < E\pi_{2F} \quad \text{if} \quad 2 < \bar{\theta} < \frac{4tV}{a} . \]
Therefore, $S_{1F} < S_{2F}$ when $2 < \bar{\theta} < \frac{4tV}{a}$. \textbf{QED.}

\textbf{Proof of Proposition 9}

Note that
\[ q^*_2 - q^*_1 = \frac{a}{2tV}(1 - r(\bar{\theta} - 1)). \]
Thus, $q^*_2 > q^*_1$ when $r < \frac{1}{\bar{\theta} - 1}$. \textbf{QED.}
References


