Impact of Network Externality in the Security Software Market*

Debabrata Dey
University of Washington, Michael G. Foster School of Business, Seattle, WA 98195, USA
ddey@uw.edu

Guoying Zhang
Midwestern State University, Dillard College of Business, Wichita Falls, TX 76308, USA
grace.zhang@mwsu.edu

February 18, 2011

Abstract

The market for security software has witnessed an unprecedented growth in recent years. A closer examination of this market reveals certain idiosyncrasies that are not observed in a traditional software market. For example, it is a highly competitive market involving several major vendors, often with a very aggressive pricing strategy adopted by new entrants. Yet, the market coverage seems to be quite low. Prior research has not attempted to explain what aspects of security software make this market deviate from the traditional ones. In this paper, we develop a quantitative model to study this market. Our model identifies the primary reason behind this behavior—that of a negative network externality effect, which pulls the market in exactly the opposite direction when compared to the positive network externality observed in traditional software markets. Using our model, we first show that a monopoly security software market has a lower market coverage, a higher price, and a higher revenue compared to a traditional monopoly market. We then extend our analysis to an oligopoly competition. There is a unique symmetric equilibrium in the oligopoly competition, and we find that the number of vendors in the security software market tends to be larger than that in a traditional market. We also consider the effect of a new entry in an oligopoly market. Next, we evaluate the strategy of vertical differentiation. We find that, while this strategy is not desired by a monopolist, it may be adopted in a duopoly competition if the cost of development is sufficiently high, although the feasible region for differentiation is more restricted than in a traditional market. Finally, we consider two extensions to the model to account for two other possible types of network externality and find that our results hold in these extended models as well. Overall, our results highlight the unique nature of the security software market, furnish rigorous explanation for several counter-intuitive observations in the real world, and provide managerial insights for vendors on market competition and product development strategies.

Keywords: Security software, network externality, negative network effect, market structure, pricing, vertical differentiation.

*We would like to thank the participants of the weekly seminar series at the Department of Marketing and International Business and the Department of Information Systems and Operations Managment, Michael G. Foster School of Business, University of Washington, Seattle, WA; this paper has benefited from their constructive comments. Draft Only; please do not quote or circulate without permission.

1 Introduction

Ever since information systems became the underlying infrastructure for modern businesses, the security of information systems has been recognized as one of the most important aspects in assuring business continuity and protecting valuable information assets. As a result, the industry of security software, along with that of security hardware and security services, has grown rapidly in response to the continually higher demand for the protection of an ever-increasing base of information technology (IT) infrastructure. Figure 1, based on Gartner reports, shows that the worldwide security software revenue has increased quite rapidly from US$6.4 billion in 2004 to about US$14.7 billion in 2009; and even a worldwide recession in 2009 did not slow this market down. The rapid growth has far exceeded all predictions about this market; for example, Latimer-Livingston and Contu (2007) predicted that the worldwide market would be about US$13.5 billion in 2011, a level that has already been achieved by 2008. IDC reports that the security software market in Asia

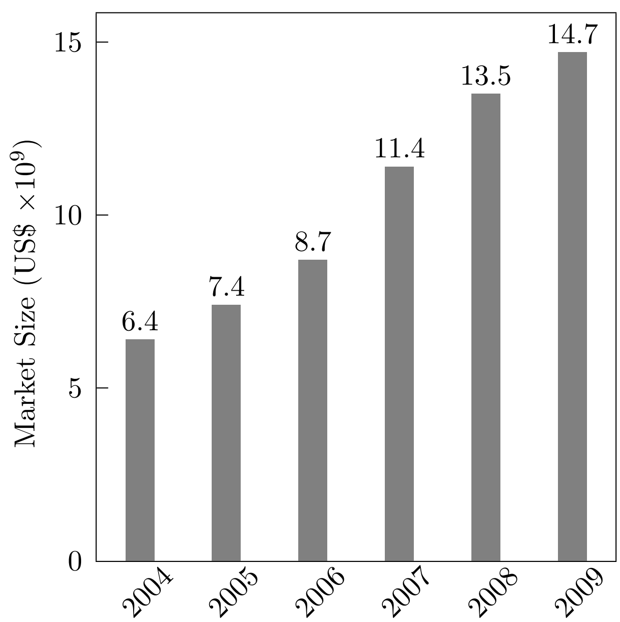


Figure 1: Growth of Worldwide Security Software Market

(excluding Japan) alone has demonstrated about 23% growth in 2005 and is keeping pace with the rest of the world in its growth (Low and Chung 2006). Security software market thus has been regarded as one of a few prominent software markets with double-digit growth rate (McCormack 2006). Understanding the nature of this market, along with its evolution and trend, is of importance to vendors as well as consumers.

In general, security software can be classified into several categories, such as antivirus software, encryption software, firewall and intrusion detection/protection systems (IDS or IPS), and spyware remover. An *antivirus software* is perhaps the most well-known type of security software, primarily used to identify and remove viruses, but can often provide protection against other malicious invasions, such as worms, phishing attacks, and Trojans. An *encryption software* is used to encrypt computer data using an encryption algorithm. With the proliferation of broadband networks, security of online data exchange becomes a key determinant to the success of E-commerce vendors. By deploying an appropriate encryption software on its Web infrastructures, an E-commerce vendor can relieve customers from the concerns on privacy and security and thus build a reputation of trust. *Firewalls* are used in local area networks to inspect the traffic going to or coming from an outside network and make decisions about whether a transmission should be allowed. Nowadays, Microsoft bundles a firewall with its Windows operating systems that can run on a personal computer. A *spyware remover* is a tool to detect and remove spyware—a piece of software that gets installed on a computer, usually through the Internet, without the user's permission. Although a spyware does not paralyze the computer or make modifications to the data, it can monitor the user's behavior, collect various types of personal information, and pass the information to the party initiating the installation of the spyware.

In a traditional off-the-shelf software market, users usually enjoy a higher network utility derived from a larger market share, which is often referred to as the positive network externality (Katz and Shapiro 1985, 1986). This positive network externality primarily arises from users' need for compatibility—the need to share files and information, to edit and critique documents created by others, and, most importantly, to work in a collaborative setting. It is well-known that a positive network externality can lead to a near-monopoly market condition: if a vendor's market share is large enough to exceed a critical mass, other competitors will lose opportunities to enter the market. Studies on the markets for off-the-shelf application software (such as Spreadsheet and Word processor) empirically validate this near-monopolistic structure (Brynjolfsson and Kemerer 1996, Liebowitz and Margolis 1999). However, the market for off-the-shelf security software is markedly different. It is characterized by many vendors, with no single dominant player. For example, the market of anti-virus software has several major players including Symantec, McAfee, Trend Micro, and Computer Associates, besides dozens of other smaller companies.[1] The total

[1] Just to name a few, consider products such as Avast, Sophos, AVG, Bitdefender, Kaspersky, Panda, Avira, G-Data, F-Secure, Webroot, ESET, Vipre, PCTools, SystemShield, SpySweeper, SystemTech, SystemWorks, Virex, and Iolo.

market share of the top 6 anti-virus software vendors is well below 50%. Many of the current anti-virus products in the market did not exist even a couple of years ago. Similar trend is observed with other security software products as well—Fosfuri and Giarratana (2004) found that, between 1989 and 1998, 270 vendors entered this market, with a very high percentage not surviving beyond two years (Giarratana 2004). Furthermore, despite the large number of existing security software vendors, a very high percentage of information systems and individual computers are still lacking in basic protection (Gordon and Loeb 2002, Lacy 2006). These observations naturally lead to several questions:

- Why are there so many vendors in the security software market?
- Why is the market coverage still low?
- Despite being so competitive and turbulent, what makes this market so attractive to new entrants?
- Is this the long-term trend or a transitory phase in market evolution?

The objective of this research is to develop a quantitative model to address these questions to obtain useful insights about the market.

It is difficult to explain the competitiveness observed in this market under the assumption of a positive network externality. In fact, from the perspective of a user of a security software, there is little additional utility to be derived from the market share of the product. A security software is simply used to prevent security exploitations, and there is hardly any benefit from the compatibility of user data. Therefore, the market of security software does not exhibit as strong a positive network effect as traditional software markets do.[2]

Instead, our analysis finds a negative network externality effect in this market. When a user adopts a security software, there are two benefits: (i) a direct benefit—representing the mitigation effect on direct security attacks by hackers, and (ii) an indirect benefit—arising from the prevention of indirect attack or infection from other users in the network (Ogut et al. 2005). In an indirect attack, a system is not a direct target, but could become an eventual target from the security exploitation of another system. Typical examples of indirect attacks include the prevalence of

[2]We should point out that, while compatibility concern is the primary force behind the positive network externality effect in traditional software markets (Gandal 1995), there may well be other contributing factors for this effect. For example, a larger market share of a product may provide additional utility to a user in terms of better product support, availability of helpful tips and resources (such as end-user forums), and better maintenance (in terms of more frequent updates and upgrades). Since these are all applicable to a security software, there may indeed be some positive network effect in this market as well. However, since the major factor (compatibility) is missing in this context, the positive network effect is significantly weaker here, when compared to the traditional software market.

Internet worms (Braverman 2005) and the wide presence of BOT net agents (Sancho 2005), which could launch large-scale attack with the ability to convert ordinary nodes into malicious agents. More than 3 million computers are thought to be part of a BOT net, with 200,000 new machines being added each month (Sancho 2005). The user's indirect benefit eventually leads to a negative network effect—the larger the market coverage of a security software, the less is the indirect benefit because the indirect threats are already mitigated, and the chance of getting infected from others reduces. Such indirect effects have also been recognized by Anderson (2001) as the "tragedy of commons," by Png et al. (2006) as the "the reason of users' inertia of taking security precautions," and by August and Tunca (2006) as an important factor in changing the users' incentive to apply security patches. Incorporation of this diminishing indirect benefit into our model leads to an increasingly less network valuation by users from a larger market coverage. We find that this negative network effect provides the primary explanation for the unique nature of the security software market.

We examine different market situations. We start by analyzing a monopoly market. We show that a monopoly security software market has lower coverage, higher price, and higher revenue compared to a traditional monopoly market (without the negative network effect). We then extend our analysis to an oligopoly competition. There is a unique symmetric equilibrium in the oligopoly market, and, as the negative externality increases, it leads to a higher price and a lower market coverage. This market is found to have more competitors in equilibrium when compared to a traditional market. We also study the effect of a new entry into the oligopoly market, and find that a new entrant is likely to adopt a more aggressive pricing strategy in this market than in a traditional market. Next, we analyze the strategy of vertical differentiation, which is a common approach adopted by software vendors (Bhargava and Choudhary 2001, Hui et al. 2007–08, Raghunathan 2000). We conclude that vertical differentiation is not an attractive strategy to a monopolist, but it may be adopted in a duopoly if the development cost is sufficiently high. However, the negative network effect significantly shrinks the region of vertical differentiation in a duopoly. Finally, we consider two extensions to the model to account for two other possible types of network externality and find that our results hold in these extended models as well. Overall, our results highlight the unique nature of the security software market, furnish explanation for several counter-intuitive observations in the real world, and provide managerial insights for vendors on market competition and product development strategies.

The rest of the paper proceeds as follows. Section 2 provides a brief review of related literature. Section 3 develops the consumer model. Sections 4 and 5 examine the market behavior under monopoly and oligopoly settings, respectively. Section 6 evaluates the strategy of vertical differentiation. We discuss two possible extensions to the basic model in Section 7. Section 8 concludes the paper and offers future research directions.

2 Literature Review

Our research is related to prior work in three main areas: (i) economics of information security, (ii) network externality, and (iii) vertical product differentiation. In this section, we provide a brief overview of some of the seminal articles in these areas.

Economics of Information Security

Gordon and Loeb (2002) consider the investment decision for information system security and develop an economic model that trades off the cost of security with the expected loss from attacks. In a subsequent empirical work, Gordon and Loeb (2006) find that this kind of economic analysis is widely used in practice for security investment decisions. Chen et al. (2005) examine the issue of investing in heterogeneous IT infrastructure by considering a diversified portfolio of platforms. They consider both positive (from compatibility and interoperability) and negative (accruing from security attacks geared towards a more popular platform) network externalities and find that diversification not only reduces the variance of security loss, but minimizes the expected loss as well. Further evidence of the benefit of diversification is provided by Böhme (2005), who finds that system diversity is preferred by an insurance issuer. Ogut et al. (2005) recognize that IT security risks borne by organizations in a networked environment are interdependent and show that this interdependence reduces an organization's incentives to invest in security technologies or to buy cyber-insurance coverage. Gal-Or and Ghose (2003) and Gordon et al. (2003) study the economic and social effects of sharing the information on security breaches among organizations. Kannan and Telang (2005) study whether a market mechanism can replace the social planner; they consider an infomediary who provides monetary rewards to vulnerability identifiers and charges his customers a subscription fee for sharing the vulnerability information. They find that, from an overall social welfare perspective, a social planner outperforms the market mechanism. Anderson (2001), Anderson and Moore (2006), and Varian (2000) look at security provision from the perspectives of underlying incentives, legal liability, and network externality. Giarratana (2004) undertakes an empirical analysis of the number of existing vendors in the security software industry. Fosfuri and Giarratana (2004) evaluate the post-entry strategies for startups in this industry. Ghose and Sundararajan (2005) analyze the bundling of different security software components. They analytically show that a mixed bundling strategy is superior to pure bundling and find preliminary empirical evidence to that effect, as well.

Network Externality

The market of technology products in presence of positive network externalities has been extensively studied (Basu et al. 2003, Economides 1996, Farrell and Saloner 1986, Katz and Shapiro 1985, 1986). The primary result from this stream of research recognizes the existence of an additional utility (to the consumer) derived from the vendor's market size, in addition to the utility derived directly from the product. Most subsequent studies on software markets make the positive network externality, and the resulting monopoly structure, a common assumption. However, as mentioned earlier, the market of security software does not exhibit a strong positive network externality. Instead, this market is unique because the user's valuation diminishes with an increasing market size. While this has not been clearly recognized as a negative externality, its effect has been noted by Ogut et al. (2005). Png et al. (2006) also identify a negative effect similar to that of public goods to explain users' inertia in taking security precautions. August and Tunca (2006) study how to motivate software users to adopt security patches and investigate the impact of a negative network externality on these policies.

Vertical Differentiation

There is a vast literature that examines the marketing strategy of vertical differentiation in general (e.g., Banker et al. 1998, Desai 2001, Gabszewicz and Thisse 1979, 1980, Gabszewicz et al. 1986, Moorthy 1984, 1988, Oren et al. 1984, Randall et al. 1998, Vanderbosch and Weinberg 1995). The typical model setup is a symmetric duopoly where two identical vendors choose quality and price levels in a multi-stage game. Researchers have also looked at the issue of differentiation and versioning in the traditional software market (Bhargava and Choudhary 2001, Raghunathan 2000). However, no special attention has been given in prior literature to the issue of vertical differentiation in the security software market.

3 The Model

Consumers (users) of security software are heterogeneous because the amount of benefit from thwarting an attack would vary from user to user. In order to capture this, consumers are indexed by a parameter u that indicates their relative expected benefit if an attack is thwarted; we assume that u is uniformly distributed over the interval $[0,1]$. The absolute expected benefit to user u from thwarting an attack can then be expressed as Lu, where L is a constant; Lu can also be viewed as a proxy for the potential loss to user u from an attack (Gordon and Loeb 2002).

As mentioned in Section 1, there are two types of benefits derived from adopting a security software—direct and indirect. First, consider the direct benefit. Assume that hackers could launch successful attacks on an unprotected system at an average rate of λ_D. Therefore, by adopting a security software, user u has a direct mitigation benefit rate of $\lambda_D Lu$.

Next, we consider the indirect benefit. Given the current level of Internet adoption and the increasing affordability of the broadband technology, users' computers are considered to be interconnected. Therefore, unprotected systems might replicate malicious codes and pass them to connected peers. At times, a hacker may attack a system indirectly, after first breaching the security of several other systems and using them as intermediate nodes to launch the attack. In other words, the existence of security software in one system can, indirectly, reduce attacks to others. Let x be the fraction of users who have adopted security software. Then, an indirect attack is possible from the $(1-x)$ unprotected fraction of users, so we model the rate of indirect attack as $\lambda_I(1-x)$, where λ_I is a base rate of indirect attack (when no user is protected). Therefore, a user adopting a security software avoids indirect attacks from the unprotected users and derives an indirect utility of $\lambda_I(1-x)Lu$. It is now obvious that a larger market share (larger x) leads to a reduction in this indirect utility. At the extreme, if all the users are equipped with security software, *no* user derives an indirect benefit from adopting the security software. This is similar to the free riding behavior in network systems and the feature of public goods in economics (Anderson 2001, Png et al. 2006).

The total benefit (per unit time) to user u from adopting the software, in a market with coverage x, can then be written as:

$$B_u = \lambda_D Lu + \lambda_I(1-x)Lu = \lambda_D Lu\left(1 + g(1-x)\right),$$

where $g = \lambda_I/\lambda_D$. Clearly, the parameter g is a proxy for the negative network externality effect—the higher the g, the larger is the potential indirect benefit and, hence, the more significant is the negative network effect. Writing the above expression in this form provides us with the flexibility to easily capture various levels of the relative indirect utility, which can be attributed to software characteristics as well as the network connectivity. For example, anti-virus and anti-spyware software have a higher indirect effect and hence a higher g, whereas an encryption software might have a lower g. A well-connected network is likely to have a higher value of g, when compared to a sparser network.

Security software products are usually licensed as a subscription for a year. Upon expiration, the user must renew the license to continue getting the service. Let P be the subscription price (per unit time). A user would adopt a security software if the total benefit from the software is larger than its subscription price: $B_u \geq P$. The marginal user u who is indifferent between adopting and not adopting the security software must then satisfy the following condition:[3]

$$\lambda_D L u \left(1 + g(1-x)\right) - P = 0.$$

As shown in Figure 2, any user to the right of this marginal user adopts the software, whereas anyone to the left does not. Therefore, $u = 1 - x$. Substituting this and letting $p = \frac{P}{\lambda_D L}$, we get:

$$p = \left(1 + g(1-x)\right)(1-x). \tag{1}$$

In other words, p in Equation (1) represents the normalized price associated with a market coverage of x. For the rest of the paper, we will use this normalized price, with appropriate subscripts, as necessary.

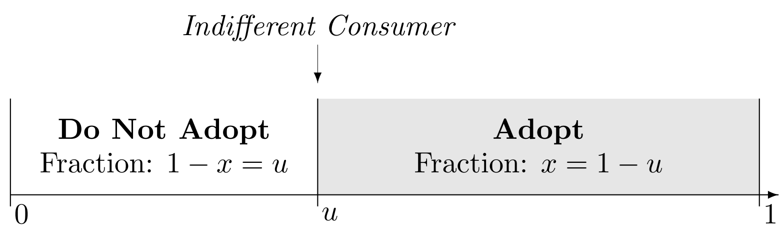


Figure 2: Consumers Choose to Adopt (or Not Adopt) Based on Their Relative Benefit

4 Monopoly Market

Existing literature on the economics of software and information goods often view a software market as a natural monopoly. We, therefore, start our analysis with the monopoly market. Let the monopoly market coverage be x_{mon}; from Equation (1), the corresponding normalized price is given by:

$$p_{\text{mon}} = (1 + g(1 - x_{\text{mon}}))(1 - x_{\text{mon}}). \tag{2}$$

Therefore, with a zero marginal cost for each user's subscription, the monopolist's objective is to select the optimal market coverage to maximize her revenue. More specifically, the monopolist solves the following optimization problem:

$$\underset{x_{\text{mon}}}{\text{Max}}\, R_{\text{mon}} = p_{\text{mon}} x_{\text{mon}} = (1 + g(1 - x_{\text{mon}}))(1 - x_{\text{mon}}) x_{\text{mon}};\ 0 \leq x_{\text{mon}} \leq 1. \tag{3}$$

[3]We should point out that, in this analysis, we are using the concept of fulfilled expected equilibrium (Katz and Shapiro 1985)—the user makes the adoption decision based on an expected value of the market size, and the realized markets size in equilibrium equals this expected value.

Solving (3), we get the following result:

PROPOSITION 1 *In a monopoly market of security software, the optimal market share and price are:*

$$x^*_{\text{mon}} = \frac{(2g+1) - \sqrt{g^2+g+1}}{3g}, \text{ and} \tag{4}$$

$$p^*_{\text{mon}} = \frac{(2g^2+2g-1) + (2g+1)\sqrt{g^2+g+1}}{9g}. \tag{5}$$

Proof: The first order condition — $\frac{dR_{\text{mon}}}{dx_{\text{mon}}} = 1 - 2x_{\text{mon}} + g(1-x_{\text{mon}})(1-3x_{\text{mon}}) = 0$— can be solved to obtain (4). Substituting (4) into (2), we get (5). Since $\frac{d^2R_{\text{mon}}}{dx^2_{\text{mon}}} = -1 - g(2-3x^*_{\text{mon}}) < 0$, the second order condition is also satisfied. ■

COROLLARY 1 *The results in Proposition 1 converge to those in a traditional monopoly market.*

Proof: For a traditional monopoly market (without the negative network effect), the revenue maximization problem is simply $\max_x x(1-x)$, where x is the market size. Therefore, the optimal price and the market size are both $\frac{1}{2}$, and the optimal revenue is $\frac{1}{4}$. We, therefore, need to prove that:

$$\lim_{g\to 0} x^*_{\text{mon}} = \frac{1}{2},\ \lim_{g\to 0} p^*_{\text{mon}} = \frac{1}{2},\ \text{and}\ \lim_{g\to 0} R_{\text{mon}}(x^*_{\text{mon}}) = \frac{1}{4}.$$

The above limits can be verified through simple algebra. ■

PROPOSITION 2 *A monopoly security software market has a smaller market coverage, higher price, and higher revenue, when compared to those in a traditional monopoly market.*

Proof: In this case, we need to show that:

$$x^*_{\text{mon}} \le \frac{1}{2},\ p^*_{\text{mon}} \ge \frac{1}{2},\ \text{and}\ R_{\text{mon}}(x^*_{\text{mon}}) \ge \frac{1}{4}.$$

First, we see that

$$x^*_{\text{mon}} - \frac{1}{2} = \frac{(2g+1) - \sqrt{g^2+g+1}}{3g} - \frac{1}{2} = \frac{(\frac{g}{2}+1) - \sqrt{g^2+g+1}}{3g}.$$

Since $\sqrt{g^2+g+1} \ge \sqrt{\frac{g^2}{4}+g+1} = \frac{g}{2}+1$, we conclude that $x^*_{\text{mon}} \le \frac{1}{2}$. Next,

$$p^*_{\text{mon}} = (1+g(1-x^*_{\text{mon}}))(1-x^*_{\text{mon}}) \ge 1 - x^*_{\text{mon}} \ge \frac{1}{2}.$$

Finally,

$$R_{\text{mon}}(x^*_{\text{mon}}) - \frac{1}{4} = p^*_{\text{mon}}x^*_{\text{mon}} - \frac{1}{4} = \frac{2(g^2+g+1)^{1.5} - (3.75g^2+3g+2)}{27g^3}.$$

Let $A = 2(g^2+g+1)^{1.5}$ and $B = 3.75g^2+3g+2$. Then $A^2 - B^2 = 4g^6 + 12g^5 + 9.9375g^4 + 5.5g^3 \ge 0$. Therefore, $A \ge B$ and $R_{\text{mon}}(x^*_{\text{mon}}) \ge \frac{1}{4}$. ■

In other words, the negative network effect makes the security software market more profitable and, hence, highly attractive to new entrants. We, therefore, turn our attention to an oligopoly market and the issue of new entry.

5 Oligopoly Market

We apply the concept of fulfilled expected Cournot equilibrium (Katz and Shapiro 1985) to study the oligopoly competition. We use the Cournot competition to model the relatively long-term decisions by vendors and use the market size as the decision variable for vendors. There are two main reasons as to why we use the Cournot competition instead of the Bertrand competition in this context:

- A large majority of subscriptions to security software happen through preloading contracts with computer manufacturers.[4] Consumers get a free trial period for the preloaded software, but must purchase the subscription if they wish to continue using it beyond the trial period. Engaging in this kind of long-term preloading agreements means that a vendor has to plan carefully about how many subscriptions it intends to sell.

- In the security software industry, timely updates to the software are quite critical—as more and more vulnerabilities are discovered over time, without timely updates, the software product gradually becomes ineffective in thwarting attacks. Thus, it is important that frequent software updates are provided online as a part of the subscription. A security software vendor must plan and commit sufficient capacity for the updating centers (e.g., servers, personnel), based on a forecasted market size. This is a long-term decision, since it is quite difficult to change the capacity of the updating center quickly.

Both these arguments point to the fact that a vendor must plan for a market size and capacity before they sell their product in the marketplace. As shown by Kreps and Scheinkman (1983), when vendors plan for a certain quantity and capacity in the first stage, a Bertrand-like price competition would still lead to a Cournot equilibrium, even if the marginal production cost is zero. Therefore, we feel that a Cournot competition provides a reasonable model of this market.

Suppose that, in equilibrium, there are n identical vendors in the market with non-negative revenue. The aggregate market size is M, where $M = \sum_{i=1}^{n} x_i$, and x_i is vendor i's market size. We

[4] For example, Dell has a contract with Symantec, McAfee, and Trend Micro to preload their anti-virus software on all the computers they sell to consumers. On the other hand, HP currently has an exclusive contract with Symantec, and rumor has it that HP will replace that with a similar contract with McAfee, once the current contract with Symantec expires.

define $\sum x_{-i}$ as the total market size of all vendors except that of vendor i, i.e., $\sum x_{-i} = M - x_i$. Extending Equation (1) for a total market coverage of M, we find that the price in this case would be $(1+g(1-M))(1-M)$, which is the valuation of the marginal user indifferent between adopting and not adopting security software. For vendor i, the revenue maximization problem can, therefore, be formulated as:

$$\underset{x_i}{\text{Max}}\, R_{\text{olig}} = \left(1 + g\left(1 - \sum x_{-i} - x_i\right)\right)\left(1 - \sum x_{-i} - x_i\right)x_i;\;\; 0 \leq \sum_{i=1}^{n} x_i \leq 1. \tag{6}$$

We can solve (6) to obtain the following result:

Proposition 3 *In an oligopoly market of security software with n identical vendors, the equilibrium market size and price for each vendor are given by:*

$$x^*_{\text{olig}} = \frac{(2g+1)(1+n) - \sqrt{4g(g+1) + (1+n)^2}}{2gn(2+n)}, \text{ and} \tag{7}$$

$$p^*_{\text{olig}} = \frac{4g(g+1) - (1+n) + (2g+1)\sqrt{4g(g+1) + (1+n)^2}}{2g(2+n)^2}. \tag{8}$$

Proof: The first order condition for (6) is:

$$\frac{\partial R_{\text{olig}}}{\partial x_i} = \left(1 - \sum x_{-i} - 2x_i\right) + g\left(1 - \sum x_{-i} - x_i\right)\left(1 - \sum x_{-i} - 3x_i\right) = 0. \tag{9}$$

This results in a quadratic equation in x_i. We solve this equation, with the restriction that $0 \leq \sum_{i=1}^{n} x_i \leq 1$, to get:

$$x_i = \frac{1 + 2g - 2g\sum x_{-i} - \sqrt{1 + g + g^2 - (g + 2g^2)\sum x_{-i} + g^2\left(\sum x_{-i}\right)^2}}{3g}. \tag{10}$$

Since the vendors are identical, we consider only the symmetric equilibrium, i.e., x_i's are all equal, as are the corresponding p_i's. Let $x_i = x_{\text{olig}}$ and $p_i = p_{\text{olig}}$. In that case, $\sum x_{-i} = (n-1)x_{\text{olig}}$. Plugging this back into (10) and solving for x_{olig}, we get (7). The equilibrium price is then given by:

$$\begin{aligned} p^*_{\text{olig}} &= \left(1 + g\left(1 - nx^*_{\text{olig}}\right)\right)\left(1 - nx^*_{\text{olig}}\right) \\ &= \frac{4g(g+1) - (1+n) + (2g+1)\sqrt{4g(g+1) + (1+n)^2}}{2g(2+n)^2}. \end{aligned}$$

Finally, we need to verify whether the second order condition is satisfied, i.e., whether:

$$\frac{\partial^2 R_{\text{olig}}}{\partial x_i^2} = -2\left(1 + gx_i + 2g\left(1 - \sum x_{-i} - 2x_i\right)\right) < 0.$$

In order to complete the proof then, we must show that $1 - \sum x_{-i} - 2x_i > 0$. Suppose this is not true, i.e., $1 - \sum x_{-i} - 2x_i \leq 0$. Since $x_i \geq 0$, this implies that $1 - \sum x_{-i} - 3x_i < 0$. Furthermore, since $\sum_{i=1}^{n} x_i \leq 1$, we have $1 - \sum x_{-i} - x_i > 0$. Therefore, the expression

$$\left(1 - \sum x_{-i} - 2x_i\right) + g\left(1 - \sum x_{-i} - x_i\right)\left(1 - \sum x_{-i} - 3x_i\right)$$

is negative, which contradicts the first order condition in (9). ■

Corollary 2 *The results in Proposition 3 converge to those in a traditional oligopoly market.*

Proof: For a traditional oligopoly market (without the negative network effect), it can be easily shown that the optimal price and the market size (for each vendor) are both $\frac{1}{n+1}$, and the optimal revenue is $\frac{1}{(n+1)^2}$. We, therefore, need to prove that:

$$\lim_{g\to 0} x^*_{\text{olig}} = \frac{1}{n+1}, \ \lim_{g\to 0} p^*_{\text{olig}} = \frac{1}{n+1}, \text{ and } \lim_{g\to 0} R_{\text{olig}}(x^*_{\text{olig}}) = \frac{1}{(n+1)^2}.$$

The above limits can be verified through algebraic manipulations. ■

Corollary 3 *The results in Proposition 3 reduce to those in Proposition 1 for $n = 1$.*

Proof: Straightforward by setting $n = 1$ in each expression. ■

Proposition 4 *The oligopoly market has the following characteristics:*

$$\frac{\partial x^*_{\text{olig}}}{\partial g} \leq 0, \ \frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0, \ \frac{\partial p^*_{\text{olig}}}{\partial g} \geq 0, \ \textit{and} \ \frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0.$$

Proof: Taking partial derivatives of x^*_{olig} and p^*_{olig} with respect to g and n, and rearranging terms, we get:

$$\begin{aligned}
\frac{\partial x^*_{\text{olig}}}{\partial g} &= \frac{2g + (1+n)^2 - (1+n)\sqrt{4g(g+1) + (1+n)^2}}{2g^2 n(2+n)\sqrt{4g(g+1) + (1+n)^2}}, \\
\frac{\partial x^*_{\text{olig}}}{\partial n} &= \frac{-(1+2g)(2+n(2+n))\sqrt{4g(1+g) + (1+n)^2} + (1+n)(2 + 8g(1+g) + n(2+n))}{2gn^2(2+n)^2\sqrt{4g(g+1) + (1+n)^2}}, \\
\frac{\partial p^*_{\text{olig}}}{\partial g} &= \frac{-(1 + 2g - 4g^2 - 8g^3 + 2n + n^2) + (1 + 4g^2 + n)\sqrt{4g(g+1) + (1+n)^2}}{2g^2(2+n)^2\sqrt{4g(g+1) + (1+n)^2}}, \text{ and} \\
\frac{\partial p^*_{\text{olig}}}{\partial n} &= \frac{-(2g+1)\left(8g(g+1) + n + n^2\right) + (-8g(g+1) + n)\sqrt{4g(g+1) + (1+n)^2}}{2g(2+n)^3\sqrt{4g(g+1) + (1+n)^2}}.
\end{aligned}$$

To find the sign of the first expression, we set $A = (1+n)\sqrt{4g(g+1) + (1+n)^2}$ and $B = 2g + (1+n)^2$. Since $A^2 - B^2 = 4g^2 n(2+n) \geq 0$, $A \leq B$ and, hence, $\frac{\partial x^*_{\text{olig}}}{\partial g} \leq 0$. To prove that the second expression is negative, let $A = (2g+1)(2 + 2n + n^2)\sqrt{4g(g+1) + (1+n)^2}$

and $B = (1+n)(2+8g(g+1)+n(2+n))$. After some algebraic manipulations, it can be shown that: $A^2 - B^2 = 4g(g+1)n^2(2+n)^2(2+4g(g+1)+n(2+n)) \geq 0$, implying $A \geq B$ and hence $\frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0$.

In order to examine the sign of the third expression, we set $A = 1+2g-4g^2-8g^3+2n+n^2$ and $B = \left(1+4g^2+n\right)\sqrt{4g(g+1)+(1+n)^2}$. It is clear that $B > 0$. Therefore, if $A \leq 0$, $B \geq A$, trivially. On the other hand, if $A > 0$, we find that: $B^2 - A^2 = 4g^2(2+n)^2\left((1+2g)^2+2n\right) \geq 0$, which implies that $B \geq A$ and $\frac{\partial p^*_{\text{olig}}}{\partial g} \geq 0$. Finally, to prove the last expression, we set: $A = (2g+1)(8g(g+1)+n+n^2)$ and $B = (-8g(g+1)+n)\sqrt{4g(g+1)+(1+n)^2}$. In this case, $A > 0$. So, if $B \leq 0$, $A > B$ holds trivially. If, however, $B > 0$, then $A^2 - B^2 = 4g(g+1)n(2+n)^3 \geq 0$, which implies that $A \geq B$ and, hence, $\frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0$. ■

Proposition 4 indicates that a higher negative network effect (i.e., a larger g) leads to a smaller market coverage and a higher price. Also, as expected, market coverage and price are decreasing with the number of vendors in the market. Next, we consider two important issues for this market—the competition level in the market and the effect of a new entry on price.

5.1 Level of Competition

As mentioned earlier, the security software market exhibits a higher level of competition when compared to the traditional software market. Based on our model, we now investigate why this is the case. Before we proceed, the following result is necessary:

Lemma 1 *The revenue for a vendor in an oligopoly market of security software has the following characteristics:* $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} \geq 0$ *and* $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial n} \leq 0$.

Proof: We know that $R_{\text{olig}}\left(x^*_{\text{olig}}\right) = p^*_{\text{olig}} x^*_{\text{olig}}$. Taking a partial derivative with respect to g and simplifying, we can write:

$$\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} = \frac{A - B}{g^3 n(2+n)^3\sqrt{4g(g+1)+(1+n)^2}},$$

where

$$\begin{aligned} A &= \left((g+1)(1+n^2)+n(2+g+2g^3)\right)\sqrt{4g(g+1)+(1+n)^2} \quad \text{and} \\ B &= (1+3g+2g^2)+n(3+4g+4g^2-2g^3-4g^4)+n^2(3+2g)+n^3(g+1). \end{aligned}$$

In order to complete the proof, we have to show $A \geq B$. If $B \leq 0$, this is obvious. Therefore, let us consider the case where $B > 0$. In that case also, $A \geq B$, since

$$A^2 - B^2 = 4g^2(g+1)n(2+n)^3(g(g+1)+n) \geq 0.$$

Lastly, we know:

$$\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial n} = p^*_{\text{olig}}\frac{\partial x^*_{\text{olig}}}{\partial n} + x^*_{\text{olig}}\frac{\partial p^*_{\text{olig}}}{\partial n}.$$

Since $p^*_{\text{olig}} > 0$, $x^*_{\text{olig}} > 0$, $\frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0$, and $\frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0$, it is clear that $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial n} \leq 0$. ■

We now try to derive the number of vendors operating in this market in equilibrium. In order to incorporate the cost of operating in the security software market, we assume that the marginal cost to the vendor for an additional software subscription is zero. This is a reasonable assumption since, once the software is developed and updating facilities are established, the marginal cost of supporting an additional subscription in terms of production, distribution, and updating is negligible. Therefore, we only consider the fixed cost incurred by vendors to develop, market, and maintain the software. Since the vendors are all identical, this cost should be the same for all vendors. We use d to denote the *normalized* fixed cost.[5] Clearly, a vendor's incentive compatibility requires her to obtain a revenue no less than this fixed cost. Therefore, if n^* denotes the number of vendors participating in the market in equilibrium, then $R_{\text{olig}}(n^*) \geq d$ and $R_{\text{olig}}(n^*+1) \leq d$. Since R_{olig} is a decreasing function of n, this implies that $n^* = \lfloor \tilde{n} \rfloor$, where $\tilde{n}$ solves:

$$R_{\text{olig}}(\tilde{n}) = p^*_{\text{olig}}(\tilde{n})x^*_{\text{olig}}(\tilde{n}) = d.$$

Combining this with Lemma 1, we can conclude that, for a given development cost d, in equilibrium, the number of vendors in an oligopoly market of security software can be no less than that in a traditional oligopoly market. In other words:

THEOREM 1 *The security software market is more competitive than the traditional market.*

Proof: The equilibrium market coverage and price for a traditional oligopoly market (without the negative network effect) are both $\frac{1}{n+1}$, resulting in a total revenue of $\frac{1}{(n+1)^2}$.

We know that:

$$\lim_{g\to 0} R_{\text{olig}}(x^*_{\text{olig}}) = \frac{1}{(n+1)^2}.$$

Furthermore, from Proposition 4, we have $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} \geq 0$. Combining these two facts, it is clear that the revenue in our case is strictly greater than that in a traditional oligopoly. Since the revenue is also a decreasing function of n (see Proposition 4), given a development cost, the number of vendors in this setting cannot be less than that in a traditional oligopoly, i.e., $n^* \geq n^*_0$, where $n^*_0 = \lim_{g\to 0} n^*$. ■

[5] Recall that we are using a normalized price in the model. Therefore, it is necessary to normalize the fixed cost in the same manner as the price. More specifically, if the absolute fixed cost (per time unit) is D, we use $d = D/(\lambda_D L)$.

This result indicates that the negative network effect in the security software market makes it more competitive. In order to illustrate this more clearly, we plot in Figure 3 how the number of vendors, n^*, changes with g. Two observations can be made from this plot. First, as expected, the number of vendors increases when the fixed cost d decreases. Second, n^* is a step-wise increasing function of g, clearly indicating that the long-run equilibrium competition level increases with the negative externality effect.

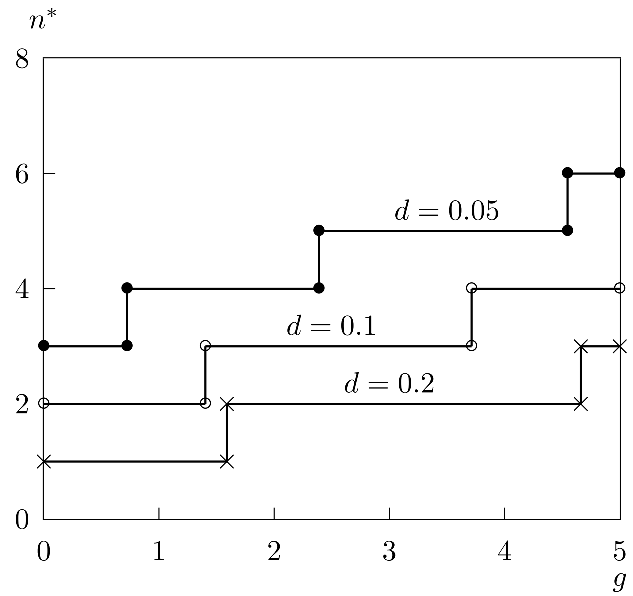


Figure 3: No. of Vendors as a Function of Negative Externality

We should mention that, even though we are considering the number of vendors in the market as an indicator for the level of competition, prior literature has proposed other metrics for this purpose. In particular, three measures are commonly used to evaluate the level of competition: Lerner index (Lerner 1934), Herfindahl-Hirschman index (Hirschman 1964), and concentration ratio (Adelman 1951). Lerner index, also known as the price-cost margin, is defined as $(p - MC)/p$, where p is the unit price and MC the marginal cost. Herfindahl-Hirschman index is defined as $\sum_{i=1}^{n}(MS_i)^2$, where n is the total number of vendors and MS_i is the market share of vendor i. Market share of a vendor, in this case, is calculated as the revenue of that vendor as a fraction of the total revenue in the market. Concentration ratio, C_k is defined as the total market share of the top k vendors in the market. For each of these the three indices, a lower value of the index implies that the market is more competitive. We can calculate these indices for the market with negative externality as well as for the traditional market without any externality ($g = 0$); they are presented in Table 1. As the marginal cost is assumed to be zero, Lerner index is trivially one and is not a meaningful

Table 1: Indices of Level of Competition

Index	Value for $g > 0$	Value for $g = 0$
Lerner Index	1	1
Herfindahl-Hirschman index	$\frac{1}{n^*}$	$\frac{1}{n_0^*}$
Concentration ratio, C_k	$\frac{k}{n^*}$	$\frac{k}{n_0^*}$

index for comparison in this context. However, the other two indices can be used. Since $n^* \geq n_0^*$, it is clear from Table 1 that the negative network externality makes the security software market more competitive, with lower Herfindahl-Hirschman index as well as lower concentration ratio.

It is also worth mentioning that, although Theorem 1, Figure 3, and Table 1 are all about the long-term equilibrium of the market, their implication must also be noted in terms of how the market is likely to behave during the transitory period—the first few years during which the industry matures. During this transitory phase, vendors typically make decisions with incomplete information about their own costs, the level of competition, and the consumers' reservation prices. A decision-maker must carefully consider the chance of success and profitability before entering a market (Gabszewicz and Thisse 1980, Van Herck 1984). If it is believed that the market would be able to support a larger number of vendors, the chance of succeeding or surviving for a new entrant increases. This, in turn, should attract more vendors to the market during the early transitory period—a result quite consistent with the empirical observation by Fosfuri and Giarratana (2004) that the security software market has attracted a very large number of vendors.

5.2 New Entry

We now turn our attention to what happens to the price when there is a new entry to the market. In 2006, when Microsoft entered the security software market, there was a large outcry about Microsoft practicing predatory pricing to drive out competition (Eckelberry 2006, Keizer 2006). Of course, it is well understood, both in theory and practice, that a new entry is supposed to drive prices down in a traditional market (Gabszewicz and Thisse 1980, Katz and Shapiro 1985, Narasimhan and Zhang 2000). The question we would like to address here is whether the existence of negative network effect makes the new entrant more aggressive and encourages her to reduce the price more drastically when compared to a traditional market.

In order to analyze this in a rigorous manner, we calculate the relative price reduction when a new entrant enters the oligopoly security software market, currently with n players, as:

$$\begin{aligned} \Delta p(n) &= \frac{p^*_{\text{olig}}(n) - p^*_{\text{olig}}(n+1)}{p^*_{\text{olig}}(n)} \\ &= 1 - \frac{(2+n)^2\left(4g(g+1) - (2+n) + (2g+1)\sqrt{4g(g+1)+(2+n)^2}\right)}{(3+n)^2\left(4g(g+1) - (1+n) + (2g+1)\sqrt{4g(g+1)+(1+n)^2}\right)}. \end{aligned} \tag{11}$$

Proposition 5 *The relative price reduction, $\Delta p(n)$, is an increasing function of g: $\frac{\partial(\Delta p(n))}{\partial g} \geq 0$, and is bounded: $\frac{1}{2+n} \leq \Delta p(n) \leq \frac{5+2n}{(3+n)^2}$.*

Proof: Let $F = \sqrt{4g(g+1)+(1+n)^2}$ and $G = \sqrt{4g(g+1)+(2+n)^2}$. Then, to prove the first part, we take partial derivative of $\Delta p(n)$ in (11) with respect to g. After some algebraic manipulations, we find that:

$$\frac{\partial(\Delta p(n))}{\partial g} = C(A-B),$$

where

$$\begin{aligned} A &= (1+2g)^3(3+2n) + F\left(13+17n+7n^2+n^3+4g(1+g)(7+6n+n^2)\right), \\ B &= G\left(2F(1+2g) + \left(6+10n+6n^2+n^3+4g(1+g)(6+4n+n^2)\right)\right), \text{ and} \\ C &= \frac{2(2+n)^2}{FG\left[(3+n)\left(1+n-4g(1+g)-F(1+2g)\right)\right]^2}. \end{aligned}$$

Since, $C > 0$, we only need to show that $A \geq B$. We find that:

$$A^2 - B^2 = 2(3+n)^2\left((2g+1)^2 + 4n(1+2g+2g^2) + 2n^2\right)(X-Y),$$

where $X = (1+n)^2 + 16g^2(1+g)^2 + 2gn^2(1+g)$ and $Y = F(1+2g)((1+n) - 4g(1+g))$. It turns out that $X^2 - Y^2 = 4g^2(1+g)^2(2+n)^4 \geq 0$, so $X \geq Y$ and, hence, $A \geq B$.

Since $\Delta p(n)$ is a monotonic function of g, to prove that $\Delta p(n)$ is bounded, simply note that $\lim_{g\to 0} \Delta p(n) = \frac{1}{2+n}$ and $\lim_{g\to\infty} \Delta p(n) = \frac{5+2n}{(3+n)^2}$. ■

It is clear from Proposition 5 that the negative externality in the security software market induces a higher reduction in price with a new entry when compared to a traditional market. If this is not taken into consideration, the entrant's pricing policy may seem "predatory," as perceived by Eckelberry (2006). Furthermore, the higher the negative externality, the higher is the extent of this price reduction, although the effect is bounded. In order to see this more clearly, we plot $\Delta p(n)$ as a function of g in Figure 4. This figure clearly illustrates that the effect of negative externality on the relative price reduction can be significant. For example, for $n = 2$, $\Delta p(n) = 25\%$ for $g = 0$, but it increases to about 35% for $g = 5$.

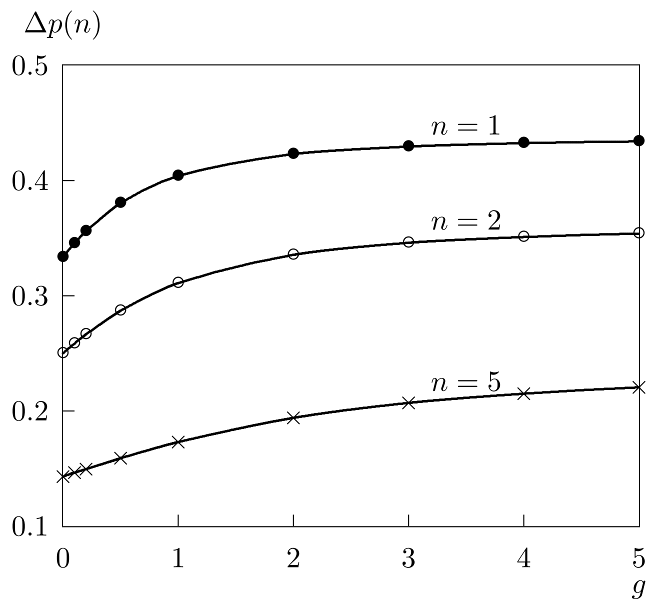


Figure 4: Relative Price Reduction as a Function of Negative Externality

6 Product Differentiation

We have shown that the market of security software is very competitive because of the negative network effect. In practice, when head-to-head market competition is fierce, vendors often resort to vertical product differentiation (Desai 2001, Gabszewicz and Thisse 1980, Moorthy 1984, 1988). Even in the near-monopolistic traditional software market, examples abound where the vendor sells several different versions of the same product (Bhargava and Choudhary 2001, Hui et al. 2007-08, Raghunathan 2000). For example, Microsoft packages basically the same Windows operating system and Office application suite differently for home and professional users. Similarly, Oracle offers Oracle 10g and Oracle 10g Express versions to target different market segments. On the other hand, in the security software market, vendors do not always offer a degraded or express version simultaneously with the full version of the product. Most of the time, the seemingly different versions are essentially different bundles of several component products. For example, Symantec offers three different bundles: Norton AntiVirus, Norton 360, and Norton 360 PremeierNorton 360 bundles anti-spyware with Norton Antivirus, and Norton 360 Premier bundles spyware and phishing protection with Norton Antivirus. Other variations of security software are often different versions based on a timeline—the yearly upgraded version is not really a simultaneous offer for a different market segment; rather, it represents continuous product improvement. This observation naturally leads to another research question: despite its highly competitive nature, why the use of vertical differentiation strategy is so limited in the security software market? In this section, we extend our model to examine whether vendors should adopt such a strategy.

We start our analysis by extending the consumer model from Section 3. Consider a market where, at the same time, two security software products are offered with similar functionalities, but different quality. The two products are characterized by a quality parameter, which can also be viewed as the effectiveness of the security software in providing the protection it is supposed to. We assume that the superior product has a quality of q_h, whereas the inferior version has a quality q_l, $0 < q_l \leq q_h \leq 1$. We model the *normalized* development cost of a product with quality level of q as cq^2, where c is a constant.

The utility of a user who adopts one of the versions of the software changes in two ways: (i) the direct utility needs to be discounted by the quality parameter $q \in \{q_l, q_h\}$, and (ii) the indirect utility also needs to be modified because now the effective coverage of each version is discounted by $q \in \{q_l, q_h\}$. The overall market characterized by $u \sim \text{Uniform}(0,1)$ can be segmented into three parts by points u_h and u_l, where $0 \leq u_l \leq u_h \leq 1$. This is represented in Figure 5. The users in

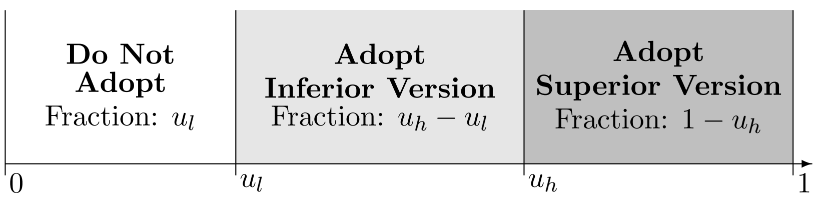


Figure 5: Segmentation of the Consumer Market

$(u_h, 1]$ choose the superior version at price p_h, the users in (u_l, u_h) choose the inferior version at price p_l, and the users in $[0, u_l)$ opt not to adopt either version. The respective market sizes for the superior and the inferior versions are: $x_h = 1 - u_h$ and $x_l = u_h - u_l$. Of course, the marginal users, u_h and u_l, must satisfy the following incentive compatibility and individual participation conditions:

$$\begin{aligned} q_l\left(1 + g(1 - q_h x_h - q_l x_l)\right) u_l - p_l &= 0, \text{ and} \\ q_h\left(1 + g(1 - q_h x_h - q_l x_l)\right) u_h - p_h &= q_l\left(1 + g(1 - q_h x_h - q_l x_l)\right) u_h - p_l. \end{aligned}$$

Substituting $u_h = 1 - x_h$ and $u_l = 1 - x_h - x_l$ into the above conditions and solving for the prices, we get:

$$p_l = q_l(1 - x_h - x_l)\left(1 + g(1 - q_h x_h - q_l x_l)\right), \text{ and} \tag{12}$$

$$p_h = \left(q_h(1 - x_h) - q_l x_l\right)\left(1 + g(1 - q_h x_h - q_l x_l)\right). \tag{13}$$

6.1 Monopoly with Differentiation

As mentioned earlier, in the software industry, it is common for a monopoly to offer products of inferior quality simultaneously with a superior-quality version (Hui et al. 2007–08, Raghunathan 2000). The purpose for such an approach is to expand the market size so as to increase the positive network externality for users, although the inferior product might cannibalize portions of the superior-quality product market. However, by now, we have established that the security software market is quite different from the traditional software market. It is, therefore, necessary to evaluate whether vertical differentiation is still a desirable strategy in the security software market.

We consider a monopolist who wants to offer two versions of a security software product characterized by quality parameter $q \in \{q_l, q_h\}$. As before, we model the development cost of quality q as cq^2. However, since the vendor is offering two versions of basically the same product, we assume that the vendor only incurs the development cost for the superior product and does not incur any additional cost for the inferior version. This makes sense since the additional production cost is negligible. The additional development cost is also minimal since the vendor can simply turn off a few of the advanced features to provide the inferior version (Raghunathan 2000).

Using (12) and (13), the total profit for the monopolist can be calculated as:

$$\begin{aligned} R_{\text{mon}} \quad &= \quad p_l x_l + p_h x_h - cq_h^2 \\ &= \left(q_l x_l(1 - 2x_h - x_l) + q_h x_h(1 - x_h)\right)\left(1 + g(1 - q_h x_h - q_l x_l)\right) - cq_h^2. \end{aligned}$$

The vendor's profit maximization problem can then be written as:

$$\begin{aligned} \underset{x_l, x_h, q_l, q_h}{\text{Max}} \quad & R_{\text{mon}} = \left(q_l x_l(1-2x_h - x_l) + q_h x_h(1-x_h)\right)\left(1 + g(1-q_h x_h - q_l x_l)\right) - cq_h^2 \\ \text{s.t.} \quad & 0 < q_l \leq q_h \leq 1, \;\; 0 \leq x_h + x_l \leq 1. \end{aligned} \tag{14}$$

Solving (14), we find:

PROPOSITION 6 *In a security software market, a monopolist would not employ a vertical differentiation strategy.*

Proof: To solve (14), we first consider the situation where the constraint $x_h + x_l \leq 1$ is binding. Then, we can substitute $x_h = 1 - x_l$ into the objective function. Of course, $q_l < q_h$; otherwise, trivially, there is no vertical differentiation. Therefore, the following first order condition must be satisfied:

$$\frac{\partial R_{\text{mon}}}{\partial q_l} = x_l(x_l - 1)(1 + g(1 - 2q_l x_l + q_h(2x_l - 1))) = 0.$$

This has three distinct roots $x_l = 0$, $x_l = 1$, and $x_l = -\frac{1+g(1-q_h)}{2g(q_h-q_l)}$. The last one violates the constraint $x_l \geq 0$ and must be discarded. The first two imply no market for one of the versions; thus, vertical differentiation should not be employed.

We now consider the situation where the constraint $x_h + x_l \leq 1$ is slack. In that case the following two first-order conditions must be satisfied: $\frac{\partial R_{\text{mon}}}{\partial x_h} = 0$ and $\frac{\partial R_{\text{mon}}}{\partial x_l} = 0$. Combining, we get:

$$q_l \frac{\partial R_{\text{mon}}}{\partial x_h} - q_h \frac{\partial R_{\text{mon}}}{\partial x_l} = 2q_l(q_h - q_l)x_l \left(1 + g(1 - q_h x_h - q_l x_l)\right) = 0.$$

This has four distinct roots: $q_l = 0$, $q_h = q_l$, $x_l = 0$, or $(1 + g(1 - q_h x_h - q_l x_l)) = 0$. Since $0 < q_l$, the first root must be discarded, so must be the last one since it implies that $q_h x_h + q_l x_l = 1 + \frac{1}{g} > 1$, which is impossible since $x_h + x_l < 1$ and $q_h, q_l \leq 1$. Therefore, either $q_h = q_l$—there is no vertical differentiation—or $x_l = 0$—there is no market for the inferior version. In either case, the monopolist does not find the vertical differentiation strategy attractive. ■

Once again, we see how the negative network effect in the security software market makes this market is different from the traditional software market, where product versioning with different prices is a common strategy to capture the marginal users. The negative externality effect in the security software market makes such a strategy sub-optimal and may partially explain why, in practice, we see little vertical product differentiation in this market, even in a monopoly situation.

6.2 Duopoly with Differentiation

We now examine whether the strategy of vertical differentiation would be adopted in a duopoly market. We use a traditional setup for vendors' differentiation choices: each vendor selects a quality level $q \in \{q_l, q_h\}$ and a price $p \in \{p_l, p_h\}$ to compete in the market, and users make rational choices of the products followed by the realization of payoffs. Furthermore, following the arguments for vertical differentiation in the monopoly market, the vendor who selects a high quality level will also be able to provide the low quality version at no additional development cost. In other words, high quality vendor can also choose to compete in the low quality market. In order to model this rigorously, let us denote by x_h the high quality product market coverage, by x_l^l the portion of the low quality vendors market coverage, and by x_l^h the portion of the high quality vendors market coverage in the low quality market. As a result, the overall market coverage is $\left(x_h + x_l^l + x_l^h\right)$.

Therefore, by rewriting Equations (12) and (13), we can obtain the prices charged in this case:

$$\begin{aligned} p_l &= q_l \left[1 - (x_h + x_l^l + x_l^h)\right] \left[1 + g\left(1 - q_h x_h - q_l\left(x_l^l + x_l^h\right)\right)\right], \text{ and} \\ p_h &= \left[q_h\left(1 - x_h\right) - q_l\left(x_l^l + x_l^h\right)\right] \left[1 + g\left(1 - q_h x_h - q_l\left(x_l^l + x_l^h\right)\right)\right]. \end{aligned}$$

The high-quality provider then solves the following optimization problem:

$$\operatorname*{Max}_{x_h, q_h} R_h = p_h x_h + p_l x_l^h - c q_h^2;\ 0 < q_h \leq 1,\ 0 < x_h + x_l^l + x_l^h \leq 1, \tag{15}$$

while the low-quality provider solves:

$$\operatorname*{Max}_{x_l, q_l} R_l = p_l x_l^l - c q_l^2;\ 0 < q_l \leq q_h,\ 0 < x_h + x_l^l + x_l^h \leq 1. \tag{16}$$

Because of the development cost asymmetry in this case, differentiation is a possible strategy, especially when the development cost is high. In order to analyze this case in a more rigorous fashion, we decompose the feasible region of the equilibrium outcome into four regions, as shown in Figure 6. It is clear that the two vendors would use vertical differentiation in Regions II and III, whereas they would not differentiate the products in Regions I and IV.

		Constraint $q_h \leq 1$	
		Binding	*Slack*
Constraint $q_l \leq q_h$	*Binding*	**Region I** $q_l = 1$ $q_h = 1$	**Region IV** $q_l = q_h$ $q_h < 1$
	Slack	**Region II** $q_l < q_h$ $q_h = 1$	**Region III** $q_l < q_h$ $q_h < 1$

Figure 6: Four Feasible Regions for Equilibrium Outcome

Lemma 2 *An equilibrium outcome cannot be in Region IV.*

Proof: We will prove this by contradiction. Let an equilibrium solution in Region IV be $q_l = q_h = q < 1$. In this region, $q_h < 1$, so the high-quality provider solves (15) without the quality constraint, and the solution must satisfy the following first-order condition:

$$\begin{aligned} \frac{\partial R_h}{\partial q_h} = x_h \Big[1 - x_h + g \Big[(1 - x_h)(1 - 2 q_h x_h) + q_l \Big(\left(2x_l^l + 3x_l^h\right) x_h \\ - x_l^l - x_l^h \left(2 - x_l^l - x_l^h\right)\Big)\Big]\Big] - 2 c q_h = 0. \end{aligned}$$

Since $q_h = q_l = q$, the two vendors should have equal market share; without loss of generality, let $x_l^h = 0$ and $x_h = x_l^l = x$. The above condition then reduces to:

$$2cq = x\left(1 - x + g\left(1 - x - qx(3 - 4x)\right)\right). \tag{17}$$

We now turn our attention to how the revenue of the low-quality provider changes with the quality of her own product; from (16):

$$\frac{\partial R_l}{\partial q_l} = x_l\left(1 - x_h - x_l^l - x_l^h\right)\left(1 + g\left(1 - q_h x_h - 2q_l\left(x_l^l - x_l^h\right)\right)\right) - 2cq_l.$$

Once again, setting $q_h = q_l = q$, $x_l^h = 0$, and $x_h = x_l^l = x$, and substituting (17), we get:

$$\left.\frac{\partial R_l}{\partial q_l}\right|_{(q,x)} = -x^2(1 + g(1 - 2qx)) < 0. \tag{18}$$

Since $q > 0$, (18) simply means that the low-quality provider can increase her profit by simply decreasing q_l from q, thereby moving into Region III. Of course, since this new solution abides by the constraint $q_l \leq q_h$, it is a valid move by the low-quality provider. Furthermore, such a move by the low-quality provider is beneficial to the high-quality provider as well; this is because

$$\left.\frac{\partial R_h}{\partial q_l}\right|_{(q,x)} = -x^2(1 + g(1 - q(1 - 4x))) < 0.$$

The last inequality holds because we have $2x \leq 1$ from the market coverage constraint, which leads to $1 - 4x \geq -1$. Therefore, $1 + q(1 - 4x) > 1 - q > 0$, since $q < 1$. Clearly then, the equilibrium outcome could not have been in Region IV. ∎

Our analysis shows that the equilibrium outcome can occur in any of the other three regions. The actual outcome depends on c, the development cost parameter. This dependence can be understood intuitively; see Figure 7. First, when c is low, i.e., the cost associated with developing a high-quality product is low, both the vendors choose the highest level of quality—$q_l = q_h = 1$; there is no vertical differentiation, and the equilibrium is observed in Region I. However, as c increases beyond a threshold (γ_1), the high cost of quality forces one of the vendors to cut down on the development cost by lowering the quality ($q_l < 1$), while the other vendor maintains the high quality level ($q_h = 1$)—the equilibrium shifts to Region II, and product differentiation is observed. As c increases further, beyond a second threshold (γ_2), the high-quality vendor is also forced to reduce the quality level ($q_h < 1$), but, as shown in Lemma 2, she always maintains a quality level higher than that of the other vendor ($q_h > q_l$); the equilibrium is observed in Region III.

It may first appear from Figure 7 that the equilibrium outcome depends only on c, and not on the negative network externality parameter, g. However, g has an important role to play in

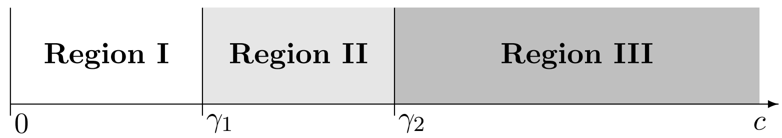


Figure 7: Equilibrium Outcome Region Changes with c

determining the equilibrium outcome and, hence, the product differentiation strategy. In order to understand the role played by g, we need to determine the threshold γ_1.

LEMMA 3 *The boundary between Regions I and II is characterized by the following threshold on c:*

$$\gamma_1 = \frac{9 + 26g + 52g^2 + 8g^3 + (4g^2 - 8g - 3)\sqrt{4g^2 + 4g + 9}}{1024g^2}. \tag{19}$$

Proof: In both the regions, $q_h = 1$; the low-quality vendor's optimization problem can, therefore, be simplified to:

$$\operatorname*{Max}_{x_l, q_l} R_l = x_l q_l \left(1 - x_h - x_l^l - x_l^h\right)\left(1 + g\left(1 - x_h - q_l\left(x_l^l + x_l^h\right)\right)\right) - c q_l^2;\ q_l > 0, x_h + x_l^l + x_l^h \leq 1.$$

The following first-order condition must be satisfied by the solution of the unconstrained problem:

$$\frac{\partial R_l}{\partial q_l} = x_l\left(1 - x_h - x_l^l - x_l^h\right)\left(1 + g\left(1 - x_h - 2q_l\left(x_l^l - x_l^h\right)\right)\right) - 2cq_l = 0.$$

Solving this, we get:

$$q_l = \frac{(1 + g(1 - x_h))\left(1 - x_h - x_l^l - x_l^h\right)x_l^l}{2\left(c + g x_l^l \left(x_l^l + x_l^h\right)\left(1 - x_h - x_l^l - x_l^h\right)\right)}.$$

Now, at the boundary of Regions I and II, $q_l = 1$. This would be the case if:

$$c = \frac{x_l^l\left(1 - x_h - x_l^l - x_l^h\right)\left(1 + g\left(1 - x_h - 2\left(x_l^l + x_l^h\right)\right)\right)}{2} = \gamma_1. \tag{20}$$

Now, if, $q_l = q_h = 1$, then the situation reduces to the oligopoly case discussed earlier with only two vendors; setting $n = 2$ in (8), we get:

$$x_h = x_l^l = x = \frac{3(2g + 1) - \sqrt{4g + 4g^2 + 9}}{16g}, \text{ and } x_l^h = 0,$$

which can be substituted into (20) to obtain (19). ■

THEOREM 2 *In a security software market with two vendors, a vertical differentiation strategy would be chosen if and only if:*

$$c > \frac{9 + 26g + 52g^2 + 8g^3 + (4g^2 - 8g - 3)\sqrt{4g^2 + 4g + 9}}{1024g^2}.$$

Proof: Vertical differentiation will be observed as long as the equilibrium outcome is not in Region I. The above condition ensures that this is indeed the case (see Lemma 3). ■

The result in Theorem 2 can be better visualized in Figure 8, where the (g, c) space is partitioned into two regions by γ_1. It is clear from Figure 8 that product differentiation decision in a duopoly

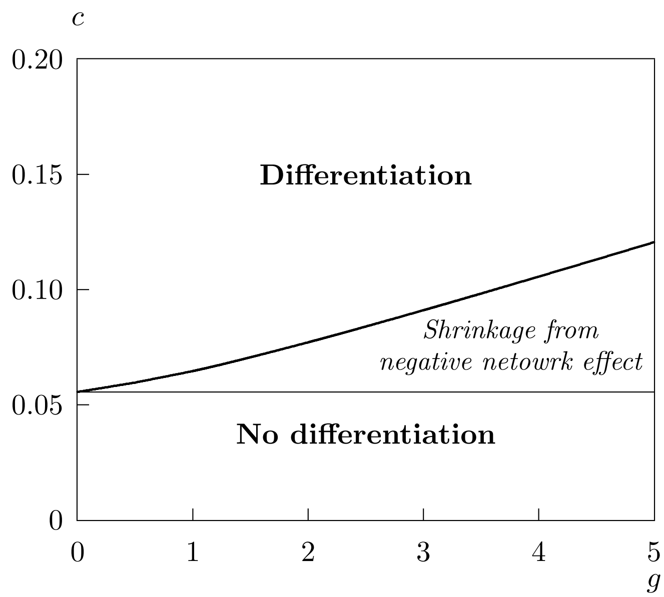


Figure 8: Product Differentiation Strategy as a Function of c and g

depends both on the development cost and the negative externality. The threshold separating the two decision choices (γ_1) is an increasing function of g. This implies that, although product differentiation is a valid strategy in the security software market, its feasible region shrinks significantly as g increases. Therefore, product differentiation is likely to be less predominant in this market when compared to a traditional market ($g = 0$). This perhaps explains why the quality levels of competing products are so close to one another in this market.

7 Rate of Attack Dependent on Market Coverage

So far, we have considered fixed rates of λ_D and λ_I for direct and indirect attacks, respectively. In this section, we will consider two extensions to this basic model by making these rates dependent on the size of the market.

The first extension incorporates a different kind of negative network externality effect in the security software market identified by Camp and Wolfram (2004), who compare this effect to that in automotive security, where attempted auto theft, in general, went down when LoJack (an auto theft response system) was introduced. In terms of information security, this can be formulated in the following manner: Hackers derive utility from breaching security in as many systems as possible. As more systems adopt security software, this utility goes down. In other words, the motivation of hackers is related to the fraction of the unprotected systems. As this number goes down, so does the motivation and, hence, the overall rate of attack. As the rate of attack decreases, the utility to a user of adopting a security software also diminishes. From now on, we will refer to this negative network effect as the *type II effect*, and denote the first effect as the *type I effect*.

The second extension, on the other hand, considers a positive network externality effect. The rationale there is as follows: as the fraction of protected users grows, hackers target the unprotected users more and more, leading to an increase in the rate of direct attack for these users. It should be noted that both these effects are recognized in the extant literature; in particular, these effects are closely related to the *choice* and *chance* models of attack on information systems (Ransbotham and Mitra, 2009). When the attacks follow a chance model, i.e., when they are opportunistic, their intensity is likely to go down with a higher level of market coverage, as hackers are also utility-maximizing agents (Ransbotham and Mitra 2009). On the other hand, if attacks follow a choice model, then they primarily target vulnerable systems, and the direct attack rate for the unprotected system should increase with a higher level of market coverage. Since, both the attack models exist in the real world, it is necessary to examine both the effects.

7.1 Type II Negative Network Effect

We incorporate the type II negative network effect by making the two attack rates a function of $(1-x)$, the fraction of unprotected systems:

$$\lambda_D = \Lambda_D(1-x)^{r-1} \text{ and } \lambda_I = \Lambda_I(1-x)^{r-1},$$

where $r \geq 1$ denotes the strength of the type II effect—as r approaches 1, this effect disappears. As before, we let $g = \lambda_D/\lambda_I = \Lambda_D/\Lambda_I$ be the parameter representing the type I effect. Through simple rearrangement of terms, Equation (1) changes to:

$$p = (1 + g(1-x))\,(1-x)^r. \tag{21}$$

Monopoly Market

The monopolist, in this case, would want to solve the following optimization problem:

$$\underset{x_{\text{mon}}}{\text{Max}}\, R_{\text{mon}} = p_{\text{mon}} x_{\text{mon}} = (1 + g(1-x_{\text{mon}}))(1-x_{\text{mon}})^r x_{\text{mon}};\; 0 \leq x_{\text{mon}} \leq 1. \tag{22}$$

We can easily solve (22) for the following result:

Proposition 7 *In a monopoly market of security software, if both type I and type II effects are present, the optimal market share and price are given by:*

$$x^*_{\text{mon}} = \frac{g(r+3)+r+1-\sqrt{(r+1)^2(g+1)^2-4g}}{2g(r+2)} \text{ and} \tag{23}$$

$$\begin{aligned} p^*_{\text{mon}} &= \left(\frac{(g-1)(r+1)+\sqrt{(r+1)^2(1+g)^2-4g}}{2g(r+2)}\right)^r \\ &\quad \times \left(\frac{g(r+1)+r+3+\sqrt{(r+1)^2(1+g)^2-4g}}{2(r+2)}\right). \end{aligned} \tag{24}$$

Proof: The first order condition:

$$\frac{dR_{\text{mon}}}{dx_{\text{mon}}} = (1-x_{\text{mon}})^{r-1}\left(1-x_{\text{mon}}(r+1)+g\left(1-x_{\text{mon}}\right)\left(1-(r+2)x_{\text{mon}}\right)\right)=0$$

can be solved to obtain x^*_{mon} in (23), which, in turn, can be substituted into the expression for p_{mon} to obtain (24). Since

$$\frac{d^2R_{\text{mon}}}{dx^2_{\text{mon}}} = (1-x_{\text{mon}})^{r-2}\left(\left(2-x_{\text{mon}}(r+1)\right)r+g(1+r)\left(1-x_{\text{mon}}\right)\left(2-(r+2)x_{\text{mon}}\right)\right)<0$$

when $x_{\text{mon}} = x^*_{\text{mon}}$, the second order condition is also satisfied. ■

Corollary 4 *The results in Proposition 7 converge to those in Proposition 1.*

Proof: This can be shown by taking limit of each expression as $r \to 1$. ■

Oligopoly Market

As before, we assume that, in equilibrium, there are n identical vendors in the market with non-negative revenue. The aggregate market size is once again M, where $M = \sum_{i=1}^{n} x_i$, and x_i is vendor i's market size. We denote $\sum x_{-i} = M - x_i$. Since the vendors are identical, the prices set by them are all equal to $(1+g(1-M))(1-M)^r$ from (21). For vendor i, the revenue maximization problem becomes:

$$\underset{x_i}{\text{Max}}\, R_{\text{olig}} = \left(1+g\left(1-\sum x_{-i}-x_i\right)\right)\left(1-\sum x_{-i}-x_i\right)^r x_i;\;\; 0 \le \sum_{i=1}^{n} x_i \le 1. \tag{25}$$

Solving (25) the following result is obtained:

Proposition 8 *In an oligopoly market of security software with n identical vendors, if both type I and type II effects are present, the equilibrium market size and price for each vendor are given by:*

$$x^*_{\text{olig}} = \frac{g(r+2n+1)+r+n-\sqrt{(g+n+r(g+1))^2-4gn}}{2gn(r+n+1)} \text{ and} \tag{26}$$

$$\begin{aligned} p^*_{\text{olig}} &= \left(\frac{g(r+1)-r-n+\sqrt{(g+n+r(1+g))^2-4gn}}{2g(r+n+1)}\right)^r \\ &\quad \times \left(\frac{g(r+1)+r+n+2+\sqrt{(g+n+r(1+g))^2-4gn}}{2(r+n+1)}\right). \end{aligned} \tag{27}$$

Proof: Let $X_i = \sum x_{-i}$. The first order condition for (25) can then be written as:

$$\frac{dR_{\text{olig}}}{dx_i} = (1 - X_i - x_i)^{r-1} (1 - X_i - x_i(r+1) + g(1 - X_i - x_i)(1 - X_i - x_i(r+2))) = 0.$$

This results in a quadratic equation in x_i, which can solved, with the following restrictions: $0 \leq \sum_{i=1}^{n} x_i \leq 1$, $x_i = x_{\text{olig}}$, and $X_i = (n-1)x_{\text{olig}}$ to obtain (26). Then, x_{olig} can be substituted into $p^*_{\text{olig}} = (1 + g(1 - nx^*_{\text{olig}}))(1 - nx^*_{\text{olig}})^r$ to obtain the equilibrium price given by (27). We now consider the second order condition. After some effort, we can show that:

$$\frac{d^2 R_{\text{olig}}}{dx_i^2} = -(1-X_i-x_i)^{r-2}(r(2-2X_i-x_i(r+1)) + g(r+1)(1-X_i-x_i)(2-2X_i-x_i(r+2))).$$

Now, since $1-X_i-x_i > 0$, $\frac{d^2 R_{\text{olig}}}{dx_i^2} < 0$ if $2-2X_i-x_i(r+2) > 0$. Suppose this is not true, i.e., $2-2X_i-x_i(r+2) \leq 0$. This also means $1-X_i-x_i(r+2) < 0$. However, that is not possible since it implies $\frac{dR_{\text{olig}}}{dx_i} < 0$—a violation of the first order condition. ■

Corollary 5 *The results in Proposition 8 converge to those in Proposition 3.*

Proof: This can be shown by taking limit of each expression as $r \to 1$. ■

Corollary 6 *The results in Proposition 8 reduce to those in Proposition 7 for $n = 1$.*

Proof: Straightforward by setting $n = 1$ in each expression. ■

Proposition 9 *The oligopoly market with both type I and type II effects has the following characteristics:*

$$\frac{\partial x^*_{\text{olig}}}{\partial g} \leq 0, \quad \frac{\partial x^*_{\text{olig}}}{\partial r} \leq 0, \quad \frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0, \quad \frac{\partial p^*_{\text{olig}}}{\partial g} \geq 0, \quad \frac{\partial p^*_{\text{olig}}}{\partial r} \leq 0, \quad \textit{and} \quad \frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0.$$

Proof: With some algebra, we can show that:

$$\frac{\partial x^*_{\text{olig}}}{\partial g} = \frac{g(n(r-1) + r(r+1)) + (r+n)^2 - (r+n)\sqrt{(g+n+r(g+1))^2 - 4gn}}{2g^2 n(r+n+1)\sqrt{(g+n+r(g+1))^2 - 4gn}}.$$

Let $A = g(n(r-1) + r(r+1)) + (r+n)^2$ and $B = (r+n)\sqrt{(g+n+r(g+1))^2 - 4gn}$. Then, $A^2 - B^2 = -4g^2 nr(r+n+1) \leq 0$ and, hence, $\frac{\partial x^*_{\text{olig}}}{\partial g} \leq 0$. Similarly, we can show that:

$$\frac{\partial x^*_{\text{olig}}}{\partial r} = \frac{-(r+n + g^2 n(r+1) + g(r+1 + n(n+r+4))) + (1-gn)\sqrt{(g+n+r(g+1))^2 - 4gn}}{2gn(r+n+1)^2\sqrt{(g+n+r(g+1))^2 - 4gn}}.$$

Let $A = r+n+g^2 n(r+1)+g(r+1+n(n+r+4))$ and $B = (1-gn)\sqrt{(g+n+r(g+1))^2 - 4gn}$. Since $A^2 - B^2 = -4g(g+1)^2 n(r+n+1) \leq 0$, $\frac{\partial x^*_{\text{olig}}}{\partial r} \leq 0$. To show that x^*_{olig} is a decreasing function of n, we note that:

$$\frac{\partial x^*_{\text{olig}}}{\partial n} = \frac{A - B}{2gn^2(r+n+1)^2\sqrt{(g+n+r(g+1))^2 - 4gn}},$$

where $A = g^2(r+1)^2(r+2n+1)+(r+n)(r+(r+n)^2+g(3n^2(r-1)+2r(r+1)^2+n(r+1)(5r-1))$ and $B = (r + (r + n)^2 + g(2n^2 + 2n(r + 1) + (r + 1)^2))\sqrt{(g + n + r(g + 1))^2 - 4gn}$. Since, $A^2 - B^2 = -4g(g + 1)n^2(r + n + 1)^2((g - n)^2 + r + 2(g + 1)(g + n)r + (g + 1)^2r^2) \leq 0$, we conclude that $\frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0$.

Let $y = 1 - nx^*_{\text{olig}} = \frac{g(r+1)-n-r+\sqrt{(g+n+r+gr)^2-4gn}}{2g(1+n+r)}$; after some algebraic manipulations, we can show that:

$$\frac{\partial y}{\partial r} = \frac{n + 1 - y + gny}{(r + n + 1)(r + n - g(r + 1) + 2gy(r + n + 1))}.$$

Since $\frac{\partial x^*_{\text{olig}}}{\partial g} \leq 0$, $\frac{\partial y}{\partial g} = -n\frac{\partial x^*_{\text{olig}}}{\partial g} \geq 0$.

The equilibrium price can now be expressed as $p^*_{\text{olig}} = (1 + gy)y^r$. Taking partial derivative w.r.t. g, we get:

$$\frac{\partial p^*_{\text{olig}}}{\partial g} = \frac{p^*_{\text{olig}}}{y}\left[\left(r + \frac{gy}{1 + gy}\right)\frac{\partial y}{\partial g} + \frac{y^2}{1 + gy}\right] \geq 0.$$

Taking partial derivative of $p^*_{\text{olig}} = (1 + gy)y^r$ w.r.t. r, we get:

$$\begin{aligned}\frac{\partial p^*_{\text{olig}}}{\partial r} &= \frac{p^*_{\text{olig}}}{y}\left[\left(r + \frac{gy}{1 + gy}\right)\frac{\partial y}{\partial r} + y\ln(y)\right] \\ &\leq \frac{p^*_{\text{olig}}}{y}\left[\left(r + \frac{gy}{1 + gy}\right)\frac{\partial y}{\partial r} - \frac{y(1-y)(11-7y+2y^2)}{6}\right],\end{aligned} \tag{28}$$

because from power series expansion of $\ln(y)$, we can show that $\ln(y) \leq -\frac{(1-y)(11-7y+2y^2)}{6}$. We can also show that y is bounded by two limits y_1 and y_2: $y_1 \leq y \leq y_2$, where

$$y_1 = \frac{g + (g + n)r + r^2}{(n + r)^2 + g(1 + n + r)} \text{ and } y_2 = \frac{gn(1 + r) + r(1 + n + r)}{n^2 + r + 2nr + r^2 + gn(1 + n + r)}.$$

Now $\left(r + \frac{gy}{1+gy}\right)\frac{\partial y}{\partial r}$ is a decreasing function of y, and achieves its maximum at $y = y_1$. On the other hand, $\frac{y(1-y)(11-7y+2y^2)}{6}$ is a concave function of y, and its minimum happens at one of the two limits, y_1 or y_2. To complete the proof, we can show, after some algebra, that:

$$\begin{aligned}\left(r + \frac{gy_1}{1 + gy_1}\right)\frac{\partial y}{\partial r}\bigg|_{y_1} - \frac{y_1(1-y_1)(11-7y_1+2y_1^2)}{6} &\leq 0, \text{ and} \\ \left(r + \frac{gy_1}{1 + gy_1}\right)\frac{\partial y}{\partial r}\bigg|_{y_1} - \frac{y_2(1-y_2)(11-7y_2+2y_2^2)}{6} &\leq 0.\end{aligned}$$

Finally, taking partial derivative of $p^*_{\text{olig}} = (1 + gy)y^r$ w.r.t. n, we get:

$$\frac{\partial p^*_{\text{olig}}}{\partial n} = \frac{p}{y}\left(r + \frac{gy}{1 + gy}\right)\frac{\partial y}{\partial n}.$$

Therefore, to complete the proof, we simply need to show that $\frac{\partial y}{\partial n} \leq 0$. We see that:

$$\frac{\partial y}{\partial n} = \frac{A - B}{2g(r + n + 1)^2\sqrt{(g + n + r(g + 1))^2 - 4gn}},$$

where $A = r+n-g(n(r-1)+(g+1)(r+1)^2)$ and $B = (1+g(r+1))\sqrt{(g+n+r(g+1))^2-4gn}$. Since, $A^2 - B^2 = -4g(g+1)r(r+n+1)^2 \leq 0$, we conclude that $\frac{\partial y}{\partial n} \leq 0$ and $\frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0$. ■

Proposition 9 is basically an extension of Proposition 4. Therefore, as before, Proposition 9 indicates that the impact of the type I negative network effect, in presence of the type II effect is similar to the impact discussed in Proposition 4—a larger g leads to a smaller market coverage and a higher price. Also, as before, market coverage and price are decreasing with the number of vendors in the market. The new result from Proposition 9 is about the impact of the type II effect. We find that a more pronounced type II effect (i.e., a larger r) would lead to a smaller market coverage as well as a smaller price.

We now consider the impact of the type II effect on the level of competition in the market. First, we need the following result:

Lemma 4 *The revenue for a vendor in an oligopoly market of security software has the following characteristics:* $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} \geq 0$, $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial r} \leq 0$, *and* $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial n} \leq 0$.

Proof: As before, let $y = 1 - nx^*_{\text{olig}}$. The revenue can then be expressed as $R_{\text{olig}}\left(x^*_{\text{olig}}\right) = \frac{1}{n}(1+gy)(1-y)y^r$. Therefore:

$$n\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} = A\frac{\partial y}{\partial g} + (1-y)y^{r+1},$$

where

$$A = \left(g(1-y)y^r + r(1-y)y^{-1+r}(1+gy) - y^r(1+gy)\right).$$

Substituting $y = \frac{g(r+1)-n-r+\sqrt{(g+n+r+gr)^2-4gn}}{2g(1+n+r)}$, we find that:

$$A = \frac{y^r(g+1)(n-1)}{2^{r-1}\left(2+g+n+r+gr-\sqrt{(g+n+r+gr)^2-4gn}\right)} \geq 0.$$

We already know that $\frac{\partial y}{\partial g} \geq 0$ (see the proof of Proposition 9). Therefore, $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial g} \geq 0$. Next, we know:

$$\frac{\partial R_{\text{olig}}(x^*_{\text{olig}})}{\partial r} = p^*_{\text{olig}}\frac{\partial x^*_{\text{olig}}}{\partial r} + x^*_{\text{olig}}\frac{\partial p^*_{\text{olig}}}{\partial r}.$$

Since $p^*_{\text{olig}} > 0$, $x^*_{\text{olig}} > 0$, $\frac{\partial x^*_{\text{olig}}}{\partial r} \leq 0$, and $\frac{\partial p^*_{\text{olig}}}{\partial r} \leq 0$, it is clear that $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial r} \leq 0$. Similarly:

$$\frac{\partial R_{\text{olig}}(x^*_{\text{olig}})}{\partial n} = p^*_{\text{olig}}\frac{\partial x^*_{\text{olig}}}{\partial n} + x^*_{\text{olig}}\frac{\partial p^*_{\text{olig}}}{\partial n}.$$

Since $p^*_{\text{olig}} > 0$, $x^*_{\text{olig}} > 0$, $\frac{\partial x^*_{\text{olig}}}{\partial n} \leq 0$, and $\frac{\partial p^*_{\text{olig}}}{\partial n} \leq 0$, it is clear that $\frac{\partial R_{\text{olig}}\left(x^*_{\text{olig}}\right)}{\partial n} \leq 0$. ■

Theorem 3 *The type I effect makes the market more competitive whereas the type II effect makes it less competitive.*

Proof: This is straightforward from Lemma 4. The type I effect makes the revenue larger and attracts more vendors, whereas the type II effect reduces the revenue and makes the market less attractive to prospective vendors. ∎

Theorem 3 provides us with two important insights. First, not all negative network effects have the same impact. It depends on how these effects manifest themselves in a specific market. In our case, the two types of effects have quite different impact in terms of the level of competition in the market. Second, most of the real-world observations from the security software market can be explained by the type I effect. However, the type II effect provides no explanation about why the market is so competitive. For the rest of the paper, we will, therefore, ignore the type II negative network effect, as its impact seems inconsequential.

7.2 Positive Network Effect

In order to capture the positive network effect due to increased direct attack on unprotected systems (or any other factor not explicitly considered), we can simply set the rate of direct attacks to $\lambda_D x$, where, x, as before, is the market coverage. The total benefit per unit time to user u then changes to:

$$B_u = \lambda_D x L u + \lambda_I (1-x) L u = \lambda_D L u \left(x + g(1-x) \right).$$

Once again, considering the marginal user who is indifferent between adopting and not adopting, we can find the normalized demand equation as:

$$p = \left(x + g(1-x) \right) (1-x). \tag{29}$$

Initially, it may appear that the entire exercise in Sections 4 and 5 must be repeated with Equation (29). However, fortunately, a simple algebraic manipulation allows us to avoid this lengthy exercise. To see this more clearly, note that Equation (29) can be rewritten as:

$$p = \left(1 + (g-1)(1-x) \right) (1-x). \tag{30}$$

Comparing Equation (30) to Equation (1), we note that they are exactly the same, the only difference being that g in Equation (1) is substituted by $(g-1)$ in Equation (30). In other words, all our algebraic manipulations are still valid, and the trend results derived earlier still hold. For example, in an oligopoly market with n identical vendors, the equilibrium market size and price for each vendor are now given by:

$$
\begin{aligned}
x^*_{\text{olig}} &= \frac{(2g-1)(1+n) - \sqrt{4g(g-1)+(1+n)^2}}{2n(g-1)(2+n)}, \text{ and} \\
p^*_{\text{olig}} &= \frac{4g(g-1) - (1+n) + (2g-1)\sqrt{4g(g-1)+(1+n)^2}}{2(g-1)(2+n)^2}.
\end{aligned}
$$

Similarly, Lemma 1, which is the basis of most of the interesting results in Section 5, also holds. In other words, our model still provides explanations for all the real-world observations about this market as long as $g > 1$. It must be noted, however, that the interpretation of the parameter g changes somewhat when we introduce the positive network effect into the model. Earlier, there was no positive network externality in the model—$g > 0$ implied the presence of negative network externality, and $g = 0$ when the network effect was absent. However, in the extended model, the absence of the network effect is represented by $g = 1$. When $g < 1$, positive network effect dominates the market, while, on the other hand, negative network effect is the dominant force when $g > 1$. In order to see this more clearly, we define $h = -\ln(g)$ as the *network externality parameter*; as shown in Figure 9, when $h = 0$, there is no network effect and the total network effect is positive (negative) if h is positive (negative).

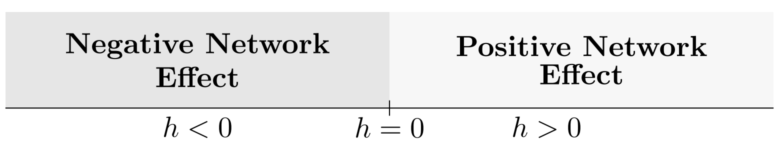
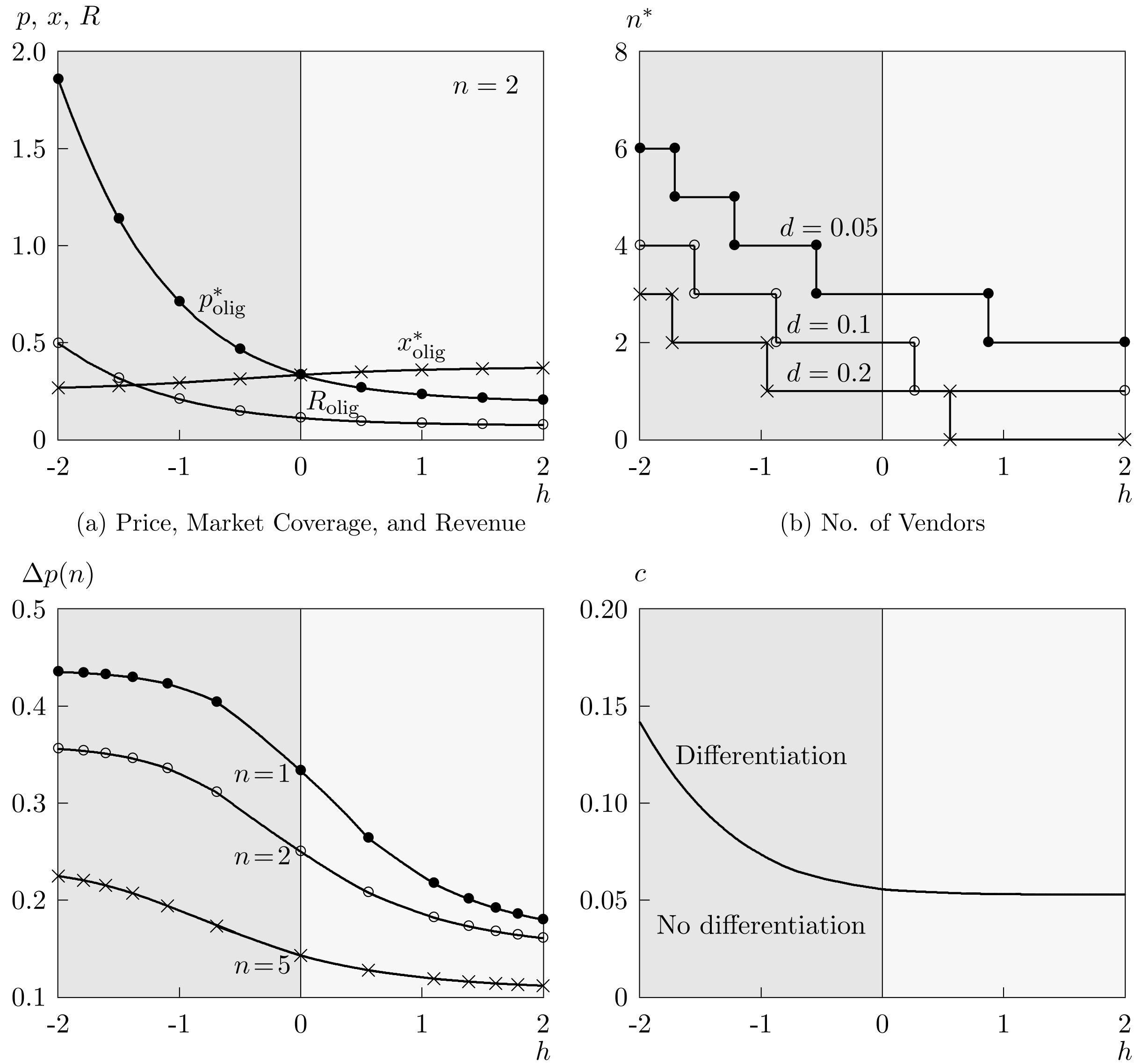


Figure 9: Market Partitions Based on Network Externality Parameter

This extended model provides a more complete picture of the market since it captures both positive and negative network effects. In Figure 10, we plot some of the interesting results in this paper as a function of h, the network externality parameter. As can be seen from figure, in the darker region ($h < 0$), our earlier results exhibit the same characteristics as before. However, in the lighter region ($h > 0$), we now have the results for the case where positive externality prevails. Not surprisingly, these results are exactly the same as what we know about traditional software markets, where positive externality has a much stronger influence. For example, as h increases, the market coverage increases, the price goes down, and the number of vendors reduces. Furthermore, the relative change in price due to a new entry into the market is much smaller when $h > 0$. Finally, while the likelihood of vertical differentiation shrinks in the presence of negative network effect, it increases slightly when $h > 0$. Viewed another way, our results not only explain the observations in



(a) Price, Market Coverage, and Revenue

(b) No. of Vendors

(c) Price Reduction

(d) Differentiation Region

Figure 10: A Complete Picture of a Market with Network Externality

the security software market correctly, but they are also consistent with observations in the market for traditional software.

8 Conclusion and Future Directions

A security software is a tool employed by individuals and organizations alike to prevent security exploitations of computerized systems. Over the last decade, the market for this type of software has seen a tremendous growth, both from the supply as well as the demand side. Unlike the traditional software market, where the supply side is dominated by only a few providers, the market for security software is highly competitive with several major players. In this paper, we study why the security software market behaves differently from the traditional one.

We find that the positive network externality enjoyed by traditional software is much weaker for security software; this is because the compatibility issue of application data across users is not a big concern. On the contrary, we find that there is a negative network externality effect for security software. This negative effect is derived from the fact that, as the market coverage grows, the chance of an indirect attack from an unprotected computer decreases. We observe that most of the differences in the prevailing market conditions of traditional and security software products arise from this negative network externality effect.

More specifically, we find that the market coverage for a security software is less when compared to a traditional market that has no negative externality, both in monopoly and oligopoly settings. We also find that the level of competition in this market is higher, and a new entrant prices more aggressively when compared to a traditional market. We also examine the strategy of vertical product differentiation. We find that this strategy would never be adopted in a monopoly market. Although, it may be adopted in a duopoly market if the cost of quality is sufficiently high, the feasible region of differentiation shrinks because of the negative network effect. Overall, our results capture the unique structure of the security software market, explain several counter-intuitive observations in practice, and provide insights for security software vendors on market competition and strategies.

Our research is the first of its kind in trying to explain the market behavior of security software providers from an analytical angle. There are several directions in which our results can be extended. For example, we do not consider the implications of public policy and governmental intervention. Since security breaches often pose a significant social impact, it is possible to argue in favor of more governmental interventions in this market. It would be interesting to study whether the social intervention is indeed desirable for the security software market. Further, in traditional software market, nonlinear pricing through volume licensing is quite common. One could investigate how nonlinear pricing can help security software vendors leverage the network effect. Another new development in the software market is the concept of software as a service, where users are not charged for the licensed copy of the software, but on the actual usage of a software module. It would be interesting to study the effect of this new service model on the security software market. Finally, vendors from traditional software markets are entering the security software market, and Internet service providers are also offering security software to their subscribers. These new competition patterns would be another interesting area of study. We are examining some of these issues in our ongoing efforts to better understand this market.

References

[1] Adelman, M.A. 1951. The Measurement of Industrial Concentration. *The Review of Economics and Statistics* **33**(4) 269–296.

[2] Anderson, J. 2001. Why Information Security is Hard—an Economic Perspective. *Proceedings of the Seventeenth Computer Security Applications Conference* 358–365.

[3] Anderson, R., T. Moore. 2006. The Economics of Information Security. *Science* **314**(5799) 610–613.

[4] August, T., T.I. Tunca. 2006. Network Software Security and User Incentives. *Management Science* **52**(11) 1703–1720.

[5] Banker, R.D., I. Khosla, K.K. Sinha. 1998. Quality and Competition. *Management Science* **44**(9) 1179–1192.

[6] Basu, A., T. Mazumdar, S.P. Raj. 2003. Indirect Network Externality Effects on Product Attributes. *Marketing Science* **22**(2) 209–221.

[7] Bhargava, H., V. Choudhary. 2001. Information Goods and Vertical Differentiation. *Journal of Management Information Systems* **18**(2) 89–106.

[8] Böhme, R. 2005. Cyber-Insurance Revisited. Presented at the *4th Workshop on the Economics of Information Security*, Cambridge, MA, June 2–3.

[9] Braverman, M. 2005. Win32/Blaster: A Case Study From Microsoft's Perspective. *Virus Bulletin Conference* 200–205. Available at: http://download.microsoft.com.

[10] Brynjolfsson, E., C. Kemerer. 1996. Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market. *Management Science* **42**(12) 1627–1647.

[11] Camp, L.J., C. Wolfram. 2004. Pricing Security. *Economics of Information Security*, L.J. Camp and S. Lewis (Eds.), Springer.

[12] Chen, P., G. Kataria, R. Krishnan. 2005. Software Diversity for Information Security. Presented at the *4th Workshop on the Economics of Information Security*, Cambridge, MA, June 2–3.

[13] Desai, P.S. 2001. Quality Segmentation in Spatial Markets: When Does Cannibalization Affect Product Line. *Marketing Science* **20**(3) 265–283.

[14] Eckelberry, A. 2006. Microsoft Practices Predatory Pricing. Sunbeltblog. Available at: http://sunbeltblog.blogspot.com/2006/06/microsoft-practices-predatory-pricing.html.

[15] Economides, N. 1996. Network Externalities, Complementarities and Invitations to Enter. *European Journal of Political Economy* **12**(2) 211–233.

[16] Farrell, J., G. Saloner. 1986. Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation. *American Economic Review* **76**(5) 940–955.

[17] Fosfuri, A., M.S. Giarratana. 2004. Product Strategies and Startups' Survival in Turbulent Industries: Evidence from the Security Software Industry. *Working Paper* Universidad Carlos III de Madrid.

[18] Gabszewicz, J.J., A. Shaked, J. Sutton, J.F. Thisse. 1986. Segmenting the Market: The Monopolist's Optimal Product Mix. *Journal of Economic Theory* **39**(2) 273–289.

[19] Gabszewicz, J.J., J.-F. Thisse. 1979. Price Competition, Quality and Income Disparities. *Journal of Economic Theory* **20**(3) 340–359.

[20] Gabszewicz, J.J., J.-F. Thisse. 1980. Entry (and Exit) in a Differentiated Industry. *Journal of Economic Theory* **22**(2) 327–338.

[21] Gal-Or, E., A. Ghose. 2003. The Economic Consequences of Sharing Security Information. Presented at the *2nd Workshop on the Economics of Information Security*, College Park, MD, May 29–30.

[22] Gandal, N. 1995. Competing Compatibility Standards and Network Externalities in the PC Software Market. *The Review of Economics and Statistics* **77**(4) 599–608.

[23] Ghose, A., A. Sundararajan. 2005. Pricing Security Software: Theory and Evidence. Presented at the *4th Workshop on the Economics of Information Security*, Cambridge, MA, June 2–3.

[24] Giarratana, S.M. 2004. The Birth of a New Industry: Entry by Start-ups and the Drivers of Vendor Growth: The Case of Encryption Software. *Research Policy* **33**(5) 787–806.

[25] Gordon, L.A., M.P. Loeb. 2002. The Economics of Information Security Investment. *ACM Transactions on Information and System Security* **5**(4) 438–457.

[26] Gordon, L.A., M.P. Loeb. 2006. Budgeting Process for Information Security Expenditures. *Communications of the ACM* **49**(10) 121–125.

[27] Gordon, L.A., M.P. Loeb, W. Lucyshyn. 2003. Sharing Information on Computer Systems: An Economic Analysis. *Journal of Accounting and Public Policy* **22**(6) 461–485.

[28] Hirschman, A.O. 1964. The Paternity of an Index. *American Economic Review* **54**(5) 761–762.

[29] Hui, W., B. Yoo, K.Y. Tam. 2007–08. The Optimal Number of Versions: Why Does Goldilocks Pricing Work for Information Goods? *Journal of Management Information Systems* **24**(3) 167–191.

[30] Kannan, K., R. Telang. 2005. Market for Software Vulnerabilities? Think Again. *Management Science* **51**(5) 726–740.

[31] Katz, M.L., C. Shapiro. 1985. Network Externalities, Competition and Compatibility. *American Economic Review* **75**(3) 424–440.

[32] Katz, M.L., C. Shapiro. 1986. Technology Adoption in the Presence of Network Externalities. *Journal of Political Economy* **94**(4) 822–841.

[33] Keizer, G. 2006. Rival Calls Microsoft's Security Pricing 'Predatory,' 'Ruthless.' Techweb Network, June 21. Available at: http://www.techweb.com/showArticle.jhtml?articleID=189600266.

[34] Kreps, D.M., J.A. Scheinkman. 1983. Quantity Precommitment and Bertrand Competition Yield Cournot Outcomes. *The Bell Journal of Economics* **14**(2) 326–337.

[35] Lacy, S. 2006. Microsoft Sweeps into Security. *BusinessWeek*, June 1. Available at: http://www.businessweek.com/technology/content/may2006/tc20060531_497986.htm.

[36] Latimer-Livingston, N.S., R. Contu. 2007. Forecast: Security Software Worldwide, 2006–2011, Update. Gartner Report. Available at: http://www.gartner.com/DisplayDocument?ref=g_search&id=510567&subref=sim.

[37] Lerner, A.P. 1934. The Concept of Monopoly and the Measurement of Monopoly Power. *The Review of Economic Studies* **1**(3) 157–175.

[38] Liebowitz, S.J., E.S. Margolis. 1999. Cause and Consequences of Market Leadership in Application Software. *Conference of Competition and Innovation in the Personal Computer Industry*, Carlsbad, California. April.

[39] Low, W., D. Chung. 2006. Asia/Pacific (Excluding Japan) Security Software 2006—2010 Forecast and Analysis. *IDC Report*. Available at: http://www.idc.com/getdoc.jsp?containerId=AP322306N.

[40] McCormack, K. 2006. Enterprise Software's Growth Pocket. (An Interview of S&P's Zaineb Bokhari by Karyn McCormack). *BusinessWeek*, May 24. Available at: http://www.businessweek.com/investor/content/may2006/pi20060524_164054.htm.

[41] Moorthy, K.S. 1984. Market Segmentation, Self-selection, and Product Line Design. *Marketing Science* **3**(4) 288–307.

[42] Moorthy, K.S. 1988. Product and Price Competition in a Duopoly. *Marketing Science* **7**(2) 141–168.

[43] Narasimhan, C., Z. Zhang. 2000. Market Entry Strategy under Firm Heterogeneity and Asymmetric Payoffs. *Marketing Science* **19**(4) 313–327.

[44] Ogut, H., N. Menon, S. Raghunathan. 2005. Cyber Insurance and IT Security Investment: Impact of Interdependent Risk. Presented at the *4th Workshop on the Economics of Information Security*, Cambridge, MA, June 2–3.

[45] Oren, S., S. Smith, R. Wilson. 1984. Pricing a Product Line. *Journal of Business* **57**(1) 73–99.

[46] Png, I.P.L., C.Q. Tang, Q. Wang. 2006. Hackers, Users, Information Security. Presented at the *5th Workshop on the Economics of Information Security*, Cambridge, UK, June 26–28.

[47] Raghunathan, S. 2000. Software Editions: An Application of Segmentation Theory to the Packaged Software Market. *Journal of Management Information Systems* **17**(1) 87–113.

[48] Randall, T., K. Ulrich, D. Reibstein. 1998. Brand Equity and Vertical Product Line Extent. *Marketing Science* **17**(4) 356–379.

[49] Ransbotham, S., S. Mitra. 2009. Choice and Chance: A Concepual Model of Paths to Information Security Compromise. *Information Systems Research* **20**(1)121–139.

[50] Sancho, D. 2005. The Future of BOT Worms. *Trend Micro White Paper*. Available at: http://us.trendmicro.com/imperia/md/content/us/pdf/threats/securitylibrary/future_of_bots_final.pdf.

[51] Van Herck, G. 1984. Entry, Exit and Profitability. *Managerial and Decision Economics* **5**(1) 25–31.

[52] Varian, H.R. 2000. Managing Online Security risks. *The New York Times*. June 1. Available at: http://www.nytimes.com/library/financial/columns/060100econ-scene.html.