Pricing Strategy and Innovation Incentives for Security Software: Theory and Evidence

Anindya Ghose Arun Sundararajan

Leonard N. Stern School of Business, New York University
aghose@stern.nyu.edu, asundara@stern.nyu.edu

Working Paper # CeDER-05-21, Center for Digital Economy Research

August 2006

We develop an analytical model that presents optimal pricing strategies for a multi-product software firm selling consumer security software. Our model highlights two aspects unique to this kind of software. The first is a supply-side effect that alters its cost structure due to the non-negligible costs of providing security updates and patches to minimize vulnerabilities and prevent security breaches. The second is a demand-side pooling effect which accrues from the fact that customers often get free substitutes for components of security software suites. In contrast with prior results on information goods, our model predicts the optimality of mixed bundling as a pricing strategy for security software vendors. The absence of these two effects leads to the optimality of pure bundling, which is consistent with past work. Another central finding of our model is that, contrary to what is commonly believed about information goods industries, the unique economics of security software actually lead to a positive relationship between bundling and innovation: the use of mixed bundling can increase a firm's incentives to develop more secure software. Our analytical model leads to many empirical predictions about the magnitude of marginal costs, the percentage discount on bundles relative to component prices and the relative demand of the bundle with respect to the demand for components. We test these propositions using a unique dataset on security software demand and pricing that we have collected from Amazon.com and find strong support for our theoretical results. Finally, we also contribute by developing and using a novel methodology based on inferring price-cost marks ups and demand elasticities, and we use it to empirically establish the optimality of mixed bundling by firms engaging in consumer security software.

Keywords: Security software, demand estimation, price discrimination, innovation, information goods, Internet, electronic commerce.

[0]We thank seminar participants at New York University, participants at the *First Annual Symposium on Statistical Challenges in E-Commerce* at the University of Maryland and participants in the *Workshop on Economics of Information Security*, for their feedback. We also thank Rong Zheng for excellent research assistance. This research was partially supported by a grant from the NET Institute.

1 Introduction

Software quality is critical to the success of the security software industry. The central driver of quality in this industry is a firm's ability to detect new vulnerabilities and deliver timely and effective solutions to its consumers. As a consequence, constant and rapid innovation by firms to provide the latest patches or updates becomes a key competency for a security software firm.

Interestingly, an informal survey of consumer shrink-wrapped software available in the market reveals that close to 80% of bundles being sold in the software market are those related to security products.[1]This points to the fact that perhaps there is something unique about the consumer security products industry which makes mixed bundling more likely or profitable in it. We conjecture that this can also have important implications for a firm's incentives to innovate and make appropriate investments in product line design and extensions. It has been argued that the IS research discipline can make significant contributions to new product development research along four dimensions: process management, project management, information and knowledge management, and collaboration and communication (Nambisan 2003). His paper's arguments apply primarily to IS support of the new product development process. discusses many areas in which the fields of IS and it discusses new product development research overlap. However, in the new digital economy firms are becoming more commonly engaged in the design and development of information technology and digital products.

In the design of information based products, it is clear that key IS research issues such as bundling (Bakos and Brynjolfsson 2000) and versioning (Shapiro and Varian 1999) are becoming more important. These product line strategies are closely linked with firm's innovation strategies. While the early literature on monopoly bundling (Adams and Yellen 1976; McAfee, McMillan and Whinston 1989) ignores its effects on entry and innovation, more recent work suggests that by deterring entry or by lowering incentives for early stage companies to enter an industry, bundling has the potential to *suppress* innovation (Bakos and Brynjolffson 1999, Nalebuff 2004). For example,

[1]Only Adobe and Microsoft are the other firms offering both bundled suites and individual components.

Microsoft has often being accused of promoting anti-competitiveness due to its bundling practices. Thus, if bundling were to result in decreased innovation through lower investments in mechanisms which enhance product quality, it could lead to a significant loss in consumer welfare.

We conjecture that while these arguments may be valid, they are less applicable to the security software industry, in which, given the emphasis on identifying vulnerability and delivering timely solutions to existing customers, innovation by *incumbents* is of paramount importance. Towards expanding on this conjecture, we presents a new model of multi-product software pricing that shows that (a) mixed bundling is uniquely optimal for security software and (b) this leads to higher incentives for innovation by the firms creating and maintaining security software. Our model highlights two aspects unique to this kind of software. The first is a supply-side effect relating to the security software update process that alters its cost structure, and the second is a demand-side effect relating to the fact that customers often get free substitutes for components of security software suites, which are bundled with products unrelated to computer security. We develop an analytical model that captures each of these effects. In contrast with prior results, this model predicts the optimality of mixed bundling as a pricing strategy; we show how removing the effects we highlight leads to the optimality of pure bundling, which is consistent with past work on information goods. The analytical model also predicts that the "discount" on a bundled software suite increases as the variable cost of high-quality updates goes up. We test the predictions of our model using data on security software demand and pricing we have gathered from Amazon.com, and find preliminary support for our theory.

As alluded to, there are a number of aspects of security software that make its economics somewhat different from those of traditional consumer software. First, the quality of security software depends less on the range of end-user features incorporated into the software, and more on the reliability and comprehensiveness of frequent updates that are critical to its effective functioning from the software vendor. For example, Symantec's AntiVirus software updates the virus definition files on each of its clients using its LiveUpdate functionality very frequently (on the order of 100 updates a year; far more than the frequency of product updates or upgrades for traditional application software). The perceived quality of this software is likely to be affected more by whether these virus definition files are developed and delivered in a timely manner, rather than by the interface or user features that AntiVirus offers. Similarly, many spam blockers (such as PanicWare's Mail-Washer) include subscriptions to spam databases that frequently update the filters on their clients; the functioning of these filters, most of which use some sort of binary classification algorithm, depends critically on periodically retraining the filters to detect new forms of spam, or updating them with newly trained ones. Since, email marketers constantly find ways to thwart existing email filters, anti-spam software vendors, in turn, create new filters intended to spot spammer's latest tricks. These filters are based on "self-learning" or "machine-learning" technologies that attempt to adapt automatically to spammers' new tricks while protecting legitimate email. Among machine-learning technologies in commercial spam filters, Bayesian filtering and neural networks are the most talked about, with Bayesian filtering generating a downright roar. In the past few months, this type of filter has been implemented in a growing number of anti-spam products, ranging from open source product SpamAssassin to an enterprise-class spam-detection module from start-up ProofPoint.

The need to continuously maintain current lists of viruses (or of spamming techniques), to develop functionality that responds to new threats, and to deliver these to each of one's customers imposes two kinds of costs on a software vendor. There is a fixed cost of maintaining the infrastructure associated with the collection of new virus lists/spamming techniques, the application development capability required to address each new threat, and the technological capability to deliver the updates. There may be economies of scope across different components of client security: the capabilities required to deliver updates can be shared, and there may be learning from countering one kind of security threat (antivirus software, for instance) that may reduce the cost of figuring out how to maintain software that counters another (personal firewalls, for example). Additionally, each new installation imposes a variable cost on the seller, since the seller has to keep track of and periodically update one additional client's computer.[2] Delivering updates and

[2]This may not be a "continuously varying" cost, but may manifest as periodic discrete increases in labor, servers and storage as one's installed base increases.

patches in a timely and efficient manner is a major challenge for most software companies. Shipping update CDs manually is extremely expensive and time consuming, while posting updates to a web site assumes that consumers and IT administrators will be able to find the update and install it properly. Within an enterprise, the challenge of delivering updates and patches is further complicated because machines are locked down and systems administrators demand control over when and how updates are applied. Software publishers are left to build update systems in-house or deal with inordinate inefficiencies and support costs associated with traditional updating methods. It would seem sensible for a software vendor to take this expected future stream of update costs into account when formulating their pricing strategy. This is especially because it is likely to be independent of the number of components of a security suite a client has installed, since the actual updates and fixes are developed independently, and the primary cost driver is likely to be simply the need to keep track of one more client.

A second major distinguishing aspect of security software is that a number of consumers may not wish to purchase certain components of a security software suite for reasons that are independent of their preferences. This is because of the nature of security software: rather than being like application software that provides consumers with software functionality, it is more like utility software that "keeps their systems running" reliably. The reason why this distinction is important is because the latter results in certain components of a security software suite being bundled with other software that is not directly related to providing security. For instance, Windows XP is sold bundled with a fairly effective personal firewall. Most consumers do not purchase Windows XP for security reasons; however, a consumer who has XP installed may not wish to pay for Norton's Firewall – this has little to do with the consumer's underlying preferences for the capabilities of the firewall. In Windows XP Service Pack 2, Internet Explorer has been updated to include pop-up ad blocking, a key feature that some competing browsers have had for months. Moreover this feature enables Internet Explorer prevent questionable Web sites from altering those windows in ways that harm the system; this innovation should put an end to the annoying proliferation of windows at some Web sites. Similarly, a user of Google's GMail is less likely to want to pay for a spam filter because GMail's spam filtering capability is quite advanced. Again, this unwillingness to pay is not on account of the user's preferences for spam (most users are likely to adopt GMail for its search, interface and storage). Other examples include SPAMfighter Standard is a free tool for Outlook and Outlook Express that automatically and efficiently filters spam and phisning fraud. Many organizations mandate the installation of antivirus software on all of their employees computers, which leads to these employees being less willing to pay for independent purchases of such software[3]. This aspect of security software manifests as a demand-side effect, wherein a random fraction of consumers have low (or no) willingness to pay for certain components, independent of their preferences for the functionality provided by this component.

We model how each of these aspects – higher costs due to security updates that vary with the number of client installations (rather than the number of components sold), and the presence of a random fraction of customers whose willingness to pay for security components is independent of their preferences – affects optimal pricing strategy for security software. We show that mixed bundling emerges as the optimal strategy, and relate the bundle discount to the magnitude of our parameters. Most importantly, we then show that this pricing strategy leads to *increased* rather than reduced incentives for innovation by the firms that sell security software. Finally, we test the predictions of our model on data we have gathered from Amazon.com, and find preliminary support for our theory.

1.1 Prior Literature

Our work is related to three streams of literature. The first is that on bundling: there is substantial literature on monopoly bundling (Adams and Yellen 1976; Schmalensee 1982; McAfee, McMillan and Whinston 1989). The bundling literature could be divided, more generally, into three broad categories, that relating to Industrial Organization such as such as analyzing efficiency (Fang and Norman 2003), entry deterrence (Nalebuff 2004a, 2004b) and profit maximization (Rochet and

[3]For instance, many consumers have Symantec's AntiVirus 8.0 installed on their computers, and therefore have a willingness to pay of zero for antivirus software; this has nothing to do with their underlying preferences for such software, and provides no information about their relative disutility from a potential computer virus infection.

Chone 1998; Manelli and Vincent 2005), that relating to bundling as a Multi-Dimensional Mechanism Design tool (Manelli and Vincent 2005) and that which includes the references on bundling of information goods (Bakos and Brynjolffson 1999, 2000, Hitt and Chen 2005). Empirical papers on bundling which span these different theoretical concepts include Crawford 2004, and Gandal et. al (2004). Gandal, et. al. (2004), provide support for existing theory about negative correlations in consumer valuations between Word Processors and spreadsheets, in the PC Office Software Markets. In this vein, our paper aims at developing new theory consistent with security products in the software market and empirically estimating theory with a unique data set.

The second stream of work is that related to software economics. Prior research in software economics has measured quality in terms of reliability and integrity of the source code, and quality models such as ISO 9126 fail to include computer security (Pfleegar 1997). Therefore software which has been certified as high quality, based on existing definitions of software quality, can have several security flaws. Siponen, Baskerville and Kuivalainena (2005) show how security techniques can be added seamlessly to agile software development methods. Several alternative security techniques have been proposed for the purposes of adding security considerations to general software development methods (Dhillon and Blackhouse 2001). Additionally, there is an impressive body of literature on software quality (for instance, Chidambaram and Kemerer (1994), Slaughter, Harter and Krishnan, 1998, 2000). Much of this literature is empirical, although our paper differs from theirs on two important dimensions. First, these tend to study quality issues for large-scale specialized software implementations in organizations, rather than the quality of mass market shrink-wrapped software. Second, they tend to assess and study software quality using supply side measures – intrinsic measures such as reliability and integrity of the source code, and the number of defects per function point – while in our model the focus is on different demand-side measures of software quality, which can be assessed by consumers. In sum, prior literature has not looked at the economic incentives of software vendors to invest in defect-free software in order to minimize security related risks. Further, the incentives for the adoption of various bundling strategies such as pure and mixed bundling and the consequences of doing so on the level of software security have not been formally studied.

Finally our work is related to growing literature on economics of information security which has looked at a variety of issues such as optimal vulnerability disclosure (Arora et al. 2004, Cavusoglu et al. 2004), how patching affects incentives of software vendors to invest in quality (Arora et al. 2005), and valuation of security technologies (Cavusoglu et al. 2002).

We have organized the rest of this paper as follows. Section 2 presents our analytical model, derives the optimal pricing strategy for security software suites, and outlines the empirical implications of our model. Section 3 describes our data and Section 4 presents the preliminary empirical results. Section 5 concludes and summarizes our ongoing work.

2 Theoretical Model

We model a multi-product monopolist who sells two security software components (for example, an antivirus program and a spam filter) for which consumer preferences and seller costs are symmetric. The results of our model generalize to directionally for multiple components.A monopolist sells two security software components, denoted I and II. The seller makes two choices: an innovation level for its software, which results in a specific level of quality, and a pricing strategy, The quality level s chosen is assumed to be identical across both components, and the fixed cost borne by the seller from it choice is denoted $F(s)$. This includes development costs associated with the components, as well as fixed costs associated with maintaining the infrastructure associated with updates at a quality level s.

The seller may price each component individually, may set a price for purchases of a bundle of both components, or may offer consumers both of these options. The price of each individual component[4] is denoted p_c, and the price of the bundle is denoted p_b. The seller's variable cost has two parts. The first part is a base cost c_0, which includes the costs of the CD, manual, packaging,

[4]Given the model's symmetry, the assumption of symmetric component quality and price is without loss of generality, and for expositional brevity, to avoid having to define multiple quality (and subsequently, price) variables that will be equal in equilibrium. We can provide a more detailed exposition on request.

shrink-wrapping and distribution. If the software is distributed digitally, this cost may be zero. The second part is a per-installation update cost c_u that represents the expected future variable costs that a seller will incur over the lifetime of the installation while keeping the installation of its software current with the latest updates (of virus definition files, for instance, as discussed in Section 1). We assume that the same cost $[c_0 + c_u]$ is incurred when the seller sells either an individual component or a bundle. That is, the marginal cost of adding the second component to a bundle containing the first is zero. This is because the base cost is largely independent of how many components are on the CD in the package, and the update cost is incurred on account of having to contact an IP address, determine the status of updates, and send updated software/data to the client: this is a per installation, rather than per component cost.

There is a unit mass of consumers. Each consumer is indexed by a parameter x which is uniformly distributed on the interval $[0, 1]$. If the quality level of the seller's software is s, the maximum utility that a consumer with index x gets from component I is $s(1 - x)$, and from component II is $s[1-(1-x)]$.[5] A fraction r of consumers own a perfect substitute for component I, and an equal fraction r of consumers own a perfect substitute for component II. As discussed in Section 1, this is on account of these components being developed and offered by sellers of software that is directly unrelated to security (firewalls with operating systems, spam filters with email clients). Since the ownership of the unrelated software does not depend on a consumer's preferences for security, the fraction is chosen randomly, and independently for each component.

The realization of these random fractions segments consumers into four groups. The first segment consists of consumers whose willingness to pay for each component is determined by their index x, and this segment is of size $(1 - r)^2$. The second and third segments consist of consumers whose willingness to pay for one of the two component is determined by their index x, and whose willingness to pay for the other component is zero. Each of these segments is of size $r(1 - r)$. The fourth segment consists of those consumers whose willingness to pay for both components is zero. This split is illustrated in Figure 1. Notice that those customers who are willing to pay for

[5] Consumers preferences for the components are therefore based on a model of these components being horizontally differentiated.

Figure 1: Illustrates consumer willingness to pay for each component as a function of their index x, in each of four segments. In each figure, the blue (thick) line denotes WTP for component I, the maroon (thin) line denotes WTP for component II, and the black (dotted) line denotes WTP for a bundle of both components.

both components all have the same willingness to pay for the bundle – this is intentional, to enable our model to capture "demand pooling" effect of larger bundles in a two-component model. Pure bundling is most effective when valuations are perfectly negatively correlated. Therefore, if mixed bundling is optimal under this scenario, it is likely to be optimal when consumer valuations are less than perfectly negative correlated.

The seller does not know which consumers are in each of these segments, and therefore, cannot directly price-discriminate based on the consumer's segment. The sequence of events is as follows: the seller chooses their quality level s; the seller chooses their pricing strategy, and the relevant values of p_c and p_B; consumers purchase either one component, both components or the bundle; each party receives its payoffs.

2.1 Pure component pricing and pure bundling

We first present some baseline results about pure component pricing, and pure bundling. Next, we characterize the optimal prices under mixed bundling, show that mixed bundling is optimal if and only if $r > 0$, how the discount offered by the seller on its security suite varies with the variable cost of security updates c_u and the fraction of customers with outside substitutes r. We then examine how the incentives to innovate increase with bundling. Finally, we discuss extensions that admit asymmetry in the demand for the individual security components, and what the model's results tell us about oligopoly.

Our first result characterizes optimal pure component pricing, when the seller does not offer a bundle of the two components.

Lemma 1 *The seller's optimal choice of pure component pricing is*

$$p_c = \frac{s + [c_o + c_u]}{2}. \tag{1}$$

All our proofs are in Appendix A. The result of this lemma is straightforward: in the absence of a bundle, the demand for each component is independent, and therefore, the seller simply chooses the profit-maximizing monopoly price for each component. Our next result characterizes optimal pricing under pure bundling, when the seller only prices its bundle, and does not sell either component independently.

Lemma 2 *The seller's optimal choice of pure bundle pricing is as follows: if* $r \leq \dfrac{s}{3s - 2[c_0 + c_u]}$, *then* $p_B = s$, *and if* $r > \dfrac{s}{3s - 2[c_0 + c_u]}$, *then* $p_B = \left(\dfrac{s + [c_0 + c_u]}{2} + \dfrac{s(1-r)}{4r}\right)$.

Lemma 2 shows that for a range of lower values of r, the seller prices the bundle at the maximum possible price s. At this price, only to those customers who are willing to pay for both products (segment 1 in Figure 1) purchase the bundle. For higher values of r, there is a sufficiently high number of customers in segments 2 and 3 to warrant lowering p_B below the maximum, in order to induce a fraction of these customers to purchase. Also, as c increases, the range of values of r over which the seller sets the maximum possible price increases, which is natural, since it is more costly to increase revenues by lowering price.

2.2 Optimal mixed bundling

Our next result specifies the seller's optimal choice of mixed bundling, and is described in Proposition 1.

Proposition 1 *The seller's optimal choice of pricing when using a mixed bundling pricing strategy is as follows:*

(a) For $r \leq \dfrac{s}{2(s-[c_0+c_u])}$*, the seller chooses the maximum bundle price:*

$$p_B = s \tag{2}$$

$$p_c = \frac{s+[c_0+c_u]}{2} + \frac{(1-r)(s-[c_0+c_u])}{2} \tag{3}$$

(b) For $r > \dfrac{s}{2(s-[c_0+c_u])}$*, the seller lowers the bundle price below the maximum, but still prices each component lower than the bundle:*

$$p_B = \frac{s+[c_0+c_u]}{2} + \frac{s}{4r} \tag{4}$$

$$p_c = \frac{s+[c_0+c_u]}{2} + \frac{s(1-r)}{4r} \tag{5}$$

Proposition 1 shows that for all values of $r < 1$, the component prices (as well as the bundle price) are *higher* than the pure components (monopoly) price derived in Lemma 1. In addition, the range of values of r over which the seller charges the maximum bundle price is *larger* under mixed bundling than under pure bundling. The intuition for this is as follows: when using mixed bundling, and with $p_B = s$, a fraction of customers in segment 1 already choose to purchase the components, since their surplus is positive form doing so. This is in contrast with their choices under pure bundling, where every customer in segment 1 bought the bundle. Moreover, the corresponding sets of customers in segments 2 and 3 also purchase the components, so the seller's lost revenue from these segments when $p_B = s$ is lower.

When $r = 0$, $p_B = p_c = s$, and the seller's strategy becomes equivalent to pure bundling, since the demand for the individual components is zero at these prices. All customers purchase the bundle, and the seller captures all their surplus. This is consistent with prior results on monopoly bundling (Bakos and Brynjolfsson, 1999) – note that while our model includes a positive variable cost $[c_0 + c_u]$, the marginal cost of adding a component to the bundle is zero. Our next proposition specifies the seller's optimal pricing strategy.

Proposition 2 *When $r > 0$, mixed bundling is the seller's optimal pricing strategy, for any $c_0 \geq 0, c_u \geq 0$.*

The proposition's proof is straightforward: it follows from the fact that $p_c < p_B$ when $r > 0$, and therefore, mixed bundling yields higher profits than any pure bundling strategy (notice that a pure bundling strategy with price p yields profits that are identical to those under $p_c = p, p_B = p$). This is an important conclusion, because it indicates that the kind of random participation that is unrelated to a customer's preferences, which we have modelled as being characteristic of consumer security software, leads to a different optimal pricing strategy.

Define the seller's discount on their bundle as the percentage difference between the price of the bundle and the sum of the prices of the individual components: that is, $[1 - (p_B/2p_c)]$. By inspection of equations (2-5), it is clear that the seller always discounts their bundle, since $p_c > (s + [c_0 + c_u])/2$, while $p_B \leq s$. The following corollary describes how this discount varies with c and r.

Corollary 1 *The discount offered by the seller on their bundled security suite is strictly increasing in the variable update cost c_u, and strictly decreasing in the fraction r of customers who have an outside substitute for a security component.*

The next corollary describes how the relative demand for the bundle and its individual components vary with c_u and r.

Corollary 2 *The demand for the bundle relative to the demand for the component increases in the update cost c_u, and decreases in r.*

Intuitively, as c increases, selling the bundle becomes increasingly desirable for the seller, since profit margins on both products shrink as c increases, but at a higher (percentage) rate for the component. A natural response for the seller is to adjust prices that shift demand away from the bundle and towards the components. Correspondingly, as r increases, the size of the segment containing potential customers for the bundle shrinks, leading to a pricing response that shifts demand in the opposite direction.

Proposition 3 *The optimal product quality under mixed bundling is always higher than that under pure component pricing. Further, for all $r > 1/4$ the optimal product quality under mixed bundling is higher than that under pure bundling.*

Proposition 3 has important implications. It highlights that bundling can actually lead to *higher* levels of investments made by firm in product innovation, which leads to higher quality for security software. Thus, unlike in other software and information goods industries, wherein the importance of innovation by new entrants is the focus, our model suggests that observing widespread bundling will actually help increase consumer welfare in the long run.

2.3 Empirical implications

Our analytical model leads to the following hypotheses about consumer security software:

- H1: If $r > 0$, the optimal pricing strategy involves mixed bundling (Proposition 2).
- H2: If the cost of updates c_u is greater then zero, then the variable cost of security software is higher than the variable cost of standard application software.
- H3: The higher the variable costs, the greater the discount offered on bundled suites of security software (Corollary 1).

- H4: The higher the variable costs, the higher the demand for the bundle relative to the demand for the components (Corollary 2).

In order to test these hypotheses, we first need estimates of the variable costs for different security software, as well as estimates of the variable costs for standard application software. We estimate the variable costs by inferring the Lerner index for each product version i (component, bundle), defined as the ratio of the markup to the price, or as $\left(\frac{p_i - c_i}{p_i}\right)$, where p_i is the retail price and c_i is the variable cost of product i. Markup is simply defined as price minus marginal cost, that is, the margin on each unit of sale. The Lerner index uses the price elasticities of demand for these products with respect to their own prices, with respect to the prices of their substitutes (the bundle for each component, and each component for the bundle), and with respect to the prices of competing products. Towards doing this estimation reliably, we follow the approach of Hausman (1994), who provides the following equation to estimate the markups for products sold by multi-product oligopolists, weighted by their market share. Since software firms generally sell multiple products and compete with multiple firms in the market, it is important to consider this formulation of the Lerner Index. Consider a set of related products indexed by i. The first-order conditions for oligopoly profit maximization yield the following system of equations:

$$s_j + \sum_{i=1}^{m} \left[\frac{p_i - c_i}{p_i} \; s_i \right] \eta_{ij} \quad = \quad 0, j = 1, 2, ..., m \tag{6}$$

Here, s_i is the demand share of product *i* **[**demand share is the ratio of revenues from product *i* to the total revenues from all related products**]**, η_{ii} is product i's elasticity of demand with respect to its own price, and η_{ij} is the cross-price elasticity of demand with respect to the price of product j.

We therefore have a system of linear equations

$$s + N'm = 0, \tag{7}$$

where s is the vector of revenue shares, $\mathbf{N}$ is the matrix of cross price elasticities $[\eta_{ij}]$, and $\mathbf{m} = [m_0, m_1, ..., m_n]$, where

$$m_i = \left(\frac{p_i - c_i}{p_i}\right) s_i \tag{8}$$

is the Lerner index of product i multiplied by its product share. The marginal costs c_i of each individual product can then be estimated by inverting $\mathbf{N}$ to solve the system of equations (7). By inverting the matrix of own and cross price elasticities, we estimate the marginal costs, c_i of each individual component of a bundle. In order to derive the matrix of own-price and cross-price elasticities, we need to estimate demand. In keeping with prior research (Ghose, Smith and Telang 2006), we estimate models of the following form:

$$\begin{aligned} Log(Rank_{pt}) = \alpha + \beta_1 Log(P_i)_{pt} + \beta_2 Log(P_j)_{pt} + \beta_3 Log(P_{Comp})_{pt} \\ + \omega X + \mu_p + \epsilon_{pt} \end{aligned} \tag{9}$$

where, *p* indexes the product (bundle or component) and *t* indexes time. The dependant variable is the Log of sales rank. The independent variables are the Amazon price (P_i) of the main product, the prices of its two competing products (P_j, and a vector of other control variables (X). Our control variables include the time since the product was released (*Datediff*), the average product rating given by customers(*Product Rating*), the number of reviewers (*Reviewers*) who have reviewed the product and a vector of Amazon's competitor prices from the marketplace (P_{Comp}). μ_p is a product fixed effect that controls for average preferences for products.[6]

Note that because of the structure of this industry, quantity and price are not jointly determined, and thus we do not face the endogeneity concerns that would normally arise in demand regressions. With regard to Amazon's own price, because software titles are produced in large quantities prior to going to market, the quantity of new products Amazon can sell is predetermined (and usually virtually infinite) at the time Amazon sets their price. This follows the standard approach taken in the literature for demand estimation of Internet product sales using the above assumed functional form for demand (Ellison and Ellison 2005, Ghose, Smith and Telang 2006).

[6] All of these are appropriate as control variables since they influence demand but do not influence the price.

3 Data and Evidence

3.1 Product Details

The 2 largest firms in the security software industry are Symantec and McAfee, whose combined revenues account for over half the total industry revenues.[7] The various products in our dataset on the security industry can categorized into the following classes.

(i) Secure Content Management: SCM is a market that reflects corporate customers' need for policy-based Internet management tools that manage Web content, messaging security, virus protection, and malicious code. (ii) Firewall/VPN Software: The firewall/VPN market consists of software that identifies and blocks access to certain applications and data. These products may also include virtual private network (VPN) encryption as an option. (iii) Intrusion Detection and Prevention Software: Intrusion detection products provide continuous monitoring of devices or networks and react to malicious activity. (iv) Security and Vulnerability Management Software: (SVM) is a comprehensive set of solutions that provides proactive alerts, and suggests fixes. They also automate the process of discovering systems on the network, identifying missing patches and installing those patches across the enterprise on a scheduled basis.

3.2 Data

To estimate the model of demand, we compiled a market-level data set on a cross-section of security software vendors encompassing the above four categories as well as regular software vendors encompassing products drawn from Business and Productivity software categories.[8] Our data are compiled from publicly available information on new product prices and sales rankings at Amazon.com. The data are gathered using automated Java scripts to access and parse HTML and XML pages downloaded from the retailer. The panel includes 68 individual software titles. These prod-

[7] There are also a few other known firms in the security software industry such as Trend Micro, Computer Associates, Veritas, Check Point and Internet Security Systems.

[8] By sifting through the list of software products available in the market, we first trivially confirmed that the majority of the bundled suites available in the regular application software market are from the Business and Productivity categories.

ucts include a mix of bundled suites as well as their individual software components, from both, security and standard application software, across different software vendors.

We collected data every 8 hours, over a 15-month period (from January 2005 to April 2006). Of the 68 products in our sample, we identify 36 as belonging to security software market and 32 belonging to the regular application software market. For each sample, we collect data on the Amazon.com sales rank (which serves as a proxy for quantity sold as described in Section 3.3), list prices, retail prices charged by Amazon.com, the date the product was released into the market, the average customer rating for the product and number of reviewers based on which the average rating was displayed. We also collected similar data from Amazon's marketplace. Note that Amazon allows even its competitors like Office Depot, CompUSA and J&R to sell used products listed as "New" or "Like new" on its marketplace. Thus, our data also takes into account some of the competitive effects on the retail prices. The summary statistics of our data are in Table 1.[9]

3.3 Demand share, revenue share and elasticity estimates

Two recent papers provided a way to map the observable Amazon.com sales rank to the corresponding number of books sold. In both cases, the authors find on a stable relationship between the ordinal sales rank of a book and the cardinal number of sales, using roughly the following Pareto relationship:

$$Log(Q_i) = K - \beta Log(R_i), \tag{10}$$

where Q_i is the demand for product i and R_i is its sales rank. Chevalier and Goolsbee (2003) calibrate this relationship using a creative and easily executed experiment where the authors obtain a book with a known number of weekly sales, purchase several copies of the book in rapid succession from Amazon.com and track the Amazon sales rank before and shortly after their purchase. Using these two points, they estimate =-0.833. Brynjolfsson, Hu and Smith (2003) calibrate this

[9]For benchmarking purposes, we also collected similar data from Buy.com: sales ranks, list prices, retail prices and so on. Similar to Amazon.com, Buy.com provides sale rankings of all of its products publicly and these sales ranks are also based on actual quantities sold at their site. The Buy.com data exhibits qualitatively similar characteristics as the Amazon.com data, and since we do not use this data further in our analysis, it is not described.

Variable	Obs	Mean	Std. Dev.	Min	Max
BSalesrank	31428	1340.14	1863.33	0	44069
C1Salesrank	31436	1012.877	1744.88	0	41766
C2Salesrank	31436	1337.01	1412.34	0	24818
BListprice	31428	442.97	443.29	59.99	1299.95
C1Listprice	31428	335.75	267.68	24.99	649
C2Listprice	31428	165.23	170.33	17.99	499
BNewprice	31428	451.77	441.51	19.97	1510.99
C1Newprice	31428	297.66	249.59	19.88	629.99
C2Newprice	31428	258.77	211.34	9.95	523.95
BComp	31428	359.32	409.90	3.65	1149.95
C1Comp	31428	208.04	220.75	.49	629.99
C2Comp	31428	157.17	165.34	.99	499
BRating	29742	3.01	1.09	1	5
C1Rating	29739	2.43	.56	1	4
C2Rating	23741	2.79	1.03	1	5
BReviewers	31118	14.64	30.86	0	190
C1Reviewers	30610	22.73	19.59	0	119
C2Reviewers	29654	17.11	34.49	0	202
Datediff	31428	338.83	223.57	-1	940

Table 1: Summary Statistics of Various Products in our Sample

relationship using data from a book publisher mapping the Amazon sales rank to the number of copies the publisher sold to Amazon. Using these data they estimate $\beta = -0.871$. Ghose and Sundararajan (2005) conduct an independent analysis for the computer software industry to convert measured sales ranks into demand data by retaining the assumption of a Pareto relationship between demand and sales rank. For this paper, we use the value of $\beta = -0.828$ and $\delta = 8.35$ that were estimated by Ghose and Sundararajan (2005) specifically for the consumer software industry. Notice that since our empirical analysis aims to compare assessed variable costs across security and non-security application software, so the absolute values of these parameters β are unlikely to change our results directionally.

Given β, we can calculate pair-wise demand and revenue shares without any knowledge of K in the following way (illustrated for one component and one bundle). Let Q_i be the demand for one product in question (for instance, a bundle), and Q_j be the demand for another (for instance, one of its components). Let their corresponding prices be p_i and p_j, and their corresponding sales ranks be R_i and R_j. Define the *demand ratio* of product i with respect to product j as:

$$\delta_{ij} = \frac{Q_i}{Q_j}, \tag{11}$$

and the *revenue ratio* of product i with respect to product j as:

$$\rho_{ij} = \frac{p_i Q_i}{p_j Q_j}. \tag{12}$$

It follows from (10) that (11) can be rewritten as:

$$\log(\delta_{ij}) = \beta[\log(R_j) - \log(R_i)], \tag{13}$$

and therefore:

$$\delta_{ij} = \left(\frac{R_j}{R_i}\right)^{\beta}. \tag{14}$$

Similarly,

$$\rho_{ij} = \frac{p_i}{p_j}\left(\frac{R_j}{R_i}\right)^{\beta}, \tag{15}$$

and therefore, for any product pair, δ_{ij} and ρ_{ij} can be inferred directly from our Amazon.com data. Thus, the demand share of product j as a fraction of the total demand for a set of products is simply:

$$\frac{Q_j}{\sum\limits_{i\in A} Q_i} = \left(1 + \sum_{i\in A, i\neq j} \delta_{ij}\right)^{-1}. \tag{16}$$

Additionally, differentiating both sides of (10) with respect to p_j and multiplying throughout by p_i yields:

$$\frac{p_i}{Q_i}\frac{dQ_i}{dp_j} = -\beta\frac{p_i}{R_i}\frac{dR_i}{dp_j}, \tag{17}$$

which in turn implies that we can directly estimate price elasticities from sales ranks without computing demand levels, so long as we have estimates of $\frac{dR_i}{dp_j}$, which we have from equation(10).

4 Analysis

We estimate these using log-linear regressions of observations over time. The results are given in tables 2, 4 and 4. On an average, we find that the own-price elasticities range between 0.78 and

1.28 for different software titles, whereas the cross-price elasticities (of a bundle with respect to its components and of the components with respect to the bundle) are significantly lower, ranging from 0.0828 to 0.877.[10]

Variable	Security SW Estimates	Application SW Estimates
$Ln(P_A)$	$1.16^{***}(0.044)$	$0.94^{***}(0.041)$
$Ln(P_{C1})$	$-0.11^{***}(0.05)$	$-0.61^{***}(0.042)$
$Ln(P_{C2})$	$-0.6^{***}(0.12)$	$-0.05^{*}(0.025)$
$Ln(Datediff)$	$1.21^{***}(0.046)$	$0.96^{***}(0.04)$
R^2	0.35	0.32

Table 2: Parameter Estimates for Bundles. The standard errors are given in parenthesis. The dependent variable is ln(sales rank of bundles). ***, ** and * denote significance at 0.01, 0.05 and 0.1 respectively.

Variable	Security SW Estimates	Application SW Estimates
$Log(P_{c1})$	$1.9^{***}(0.25)$	$1.06^{***}(0.052)$
$Log(P_B)$	-0.09$^{**}(0.04)$	$-0.22^{***}(0.038)$
$Log(P_{c2})$	$-0.15^{***}(0.03)$	$-0.2(0.012)$
$Log(Datediff)$	$0.96^{***}(0.04)$	$0.57^{***}(0.024)$
R^2	0.33	0.51

Table 3: Parameter Estimates for Component 1. The standard errors are given in parenthesis. The dependent variable is ln(sales rank of component 1). ***, ** and * denote significance at 0.01, 0.05 and 0.1 respectively.

Variable	Security SW Estimates	Application SW Estimates
$Log(P_{c2})$	$1.55^{***}(0.15)$	$1.43^{***}(0.029)$
$Log(P_B)$	$-1.06^{***}(0.28)$	$-0.99^{***}(0.16)$
$Log(P_{c1})$	$-0.84^{***}(0.055)$	$-0.66^{***}(0.04)$
$Log(Datediff)$	$0.94^{***}(0.039)$	$0.49^{***}(0.025)$
R^2	0.17	0.29

Table 4: Parameter Estimates for Component 2. The standard errors are given in parenthesis. The dependent variable is ln(sales rank of component 2). ***, ** and * denote significance at 0.01, 0.05 and 0.1 respectively.

[10]Recall that in order to impute the actual price elasticities the coefficients have to be transformed using the Pareto mapping parameter. So for example, a coefficient of 0.94 implies a elasticity of (0.94*0.828)=0.78.)

We point out that the parameters of interest (bundle and component prices) are precisely estimated and the price elasticity parameter estimates and associated standard errors are quite stable across specifications, suggesting that the estimates are robust. The other control variables suggest that, as expected, sales of new products decrease over time.

4.1 Optimality of Bundling

First, we confirm trivially by observation that each component of the suite is also sold individually, and therefore, the firms are using mixed bundling. In order to estimate the optimality of mixed bundling, we assess the extent to which each firm's profits would increase by dropping (a) its bundle (that is, offering pure component pricing), and (b) sales of its individual components (that is, offering just a pure bundle).

A multi-product firm faces two consequences of extending its product line: a "cannibalization effect" and a "revenue generating effect". The former effect captures the encroachment of additional products introduced by the firm on the market share of the firm's existing products. The latter effect captures the fact that introducing a new product expands the firm's entire market: either by bringing in new customers who were unserved by existing product lines (the "market expansion" effect) or by poaching its competitors' customers (the "business stealing" effect). The interplay between these effects eventually determines the optimality of extending one's product line. In our context, a bundle of two or more components can lead to the same effects. It can potentially cannibalize the market share of the individual components but, at the same time, it can also expand the firm's total market share and revenues.

Our first hypothesis thus aims to test the optimality of mixed bundling for security software products. To estimate the impact of bundles on individual components and vice-versa, we note that from the definition of elasticity, due to competition from individual components, the change in bundled suite sales, ΔQ_{b_i} should be given by:

$$\Delta Q_{b_i} = Q_{b_i}\ \eta_{b_i c_i} \left(\frac{P_{b_i} - P_{c_i}}{P_{b_i}} \right), \tag{18}$$

where Q_{b_i} is the total number of bundled suites of product i sold per year, η_{bc_i} is the cross price elasticity of the bundle with respect to prices of the components $c_1, c_2, .., c_i$, and the fourth term corresponds to the average discount of component prices with respect to the price of the bundled suite. Using our cross price elasticity estimates of a bundle with respect to a component product, we compute the number of bundle sales lost due to the concurrent availability of individual components. Similarly, using the elasticity estimates of a component with respect to the bundle, we compute the number of component sales lost, ΔQ_{c_i} due to the presence of a bundle.

$$\Delta Q_{c_i} = Q_{c_i}\ \eta_{c_i b_i} \left(\frac{P_{c_i} - \frac{P_{b_i}}{i}}{P_{b_i}} \right). \tag{19}$$

Suppose we want to show that the revenue gain from having component i is more than the loss due to some units of the bundle being cannibalized by the component. We know that the total revenues from component i is $P_{c_i}\ Q_{c_i}$. Conversely, the revenue loss from some units of the bundle being cannibalized is given by $\Delta Q_{b_i}\ P_{b_i}$. We need to show that $P_{c_i}\ Q_{c_i} > \Delta Q_{b_i}\ P_{b_i}$. From the equation above, $\Delta Q_{b_i}\ P_{b_i} = Q_{b_i}\ \eta_{b_i c_i}(P_{b_i} - P_{c_i})$. Hence, we need to show that

$$\begin{aligned}
P_{c_i}\ Q_{c_i} &> Q_{b_i}\eta_{b_i c_i}(P_{b_i} - P_{c_i}) \Longleftrightarrow \\
LogP_{c_i} + LogQ_{c_i} &> LogQ_{b_i} + Log\eta_{b_i c_i} + Log(P_{b_i} - P_{c_i}) \\
\beta Log(\frac{R_{b_i}}{R_{c_i}}) &> Log\ \left(\eta_{b_i c_i}(\frac{P_{b_i} - P_{c_i}}{P_{b_i}}) \right).
\end{aligned} \tag{20}$$

We are thus able to derive the “cannibalization effect” and a “revenue generating effect of each product, both bundles and individual components. While this does not take into account price changes that the firm would implement if it were to change its product line, a negative result would indicate that mixed bundling is not optimal (and therefore, while not sufficient to establish the optimality of mixed bundling empirically, it seems necessary).

As an example, we conduct this analysis for three product lines in our sample: Adobe, Norton and McAfee. For the application software family, our analysis reveals that 19.73% and 13.5% of bundle revenues were cannibalized by the individual components 1 and 2, respectively. Similarly, the bundled suite cannibalized 19.5% and 3% of component 1 and 2 revenues, respectively. In the same time period, the ratio of the total revenues of the components to that of the bundle, was approximately 34.3. Thus, given the information on prices that we have from the data, our analysis revealed that the gain in revenues from a larger market share (across all three products), more than compensated for the loss due to cannibalization, both due to the bundle as well as the individual components. For the security software family, our estimates highlight that 8.1% and 1.5 % of the bundle revenues were cannibalized by two the individual components, respectively. Similarly, the bundled suite cannibalized 9% and 2% of the component revenues, respectively. In the same time period, the ratio of the total revenues of the individual components to that of the bundle, was approximately 15.1. Again, given the information on prices that we have from the data, our analysis revealed that the gain in revenues from a larger market share (across all three products), compensated for the loss due to cannibalization, both due to the bundle as well as the individual components of the bundle. We obtain similar trends for Norton products. In principle, we could do the same analysis for the entire security and regular application software categories.

A detailed analysis of the methodology to determine the optimality of mixed bundling is given in the Appendix, where we outline the function $G'(P)$. Note that in order to do this we need to have estimates of the own-price and cross-price elasticities of the various bundles and their components which we obtain from equation (10). The results are listed in table 5 which shows that for both security software and regular application software $G'(P) > 0$. This analysis combined with that above, provides support for our hypothesis that mixed bundling is indeed optimal for software products that have positive marginal costs and this extent of profitability actually increases with increase in the marginal costs. This analysis confirms Hypothesis 1. Next, based on (11-16), we compute demand and revenue shares for each of our products, which are then used in (6) to compute the markups and the marginal costs for each of the bundled suites. These are also summarized in table 5.

Our empirical analysis confirms that on average the marginal costs for bundled suites of security software are higher than those of bundled suites of regular application software, which provides support for Hypothesis 2. Specifically, when the when the wholesale price of the manufacturer is discounted 30% from the retail price, the marginal cost of the bundled site of security software is $ 2.2 higher than that of the regular software.

We also find support for the fact that discount on bundled suites relative to components is higher for products with a higher variable cost, lending support for Hypothesis 3. That is, the discount on bundled suites of security software is higher than discount on bundles of regular application software. Specifically, when the wholesale price of the manufacturer is discounted 30% from the retail price, then there is a 20% discount on the bundled suites of security software compared to only a 1% discount on the bundled suite of regular application software.

Finally, we also compare the demand for the bundle relative to the demand for the components for security products (that is products with a higher variable cost) with regular application software products (products with lower variable costs). The estimated relative demand δ_{ij}, is marginally higher for security software compared to regular software. This implies Hypotheses 4 is also supported.

Note that in order to infer the marginal costs precisely, we needed information on prices at which these products are sold by manufacturers to retailers like Amazon. In order to get this information we collected data consisting of the average wholesale prices at which consumer retail software is sold by software manufacturers to retailers. The data was collected from New York University's computer store and consists of the average wholesale prices of approximately 55 products drawn from a similar sample of products as in our dataset. The data reveals that the average retail price is marked up by 30% from the wholesale price. In table 5 we present the analysis.

Name	Marginal Cost	Markup	Discount	Demand Ratio	G'(P)
Security S/W Bundle	11.6	0.90	0.2	0.52	0.0012
Application S/W Bundle	9.4	0.97	0.011	0.5	0.0018

Table 5: Results on marginal costs, mark-ups, bundle discounts and demand ratios, where 'B' is Bundle and 'C' is Component. The mark ups and marginal costs were imputed based on the fact that wholesale prices were discounted 30% from the retail price. G'(P)> 0 implies mixed bundling is optimal based on the method outlined in the Appendix.

5 Conclusion

This paper has presented the first empirical study of bundling in the software industry. The contributions of this study are summarized below:

(1) We have presented a new theory which relates a security firm's pricing strategy to two important aspects that distinguish security software, and have shown how these aspects affect optimal bundle and component pricing. We have demonstrated that these results are stronger for security software compared to application software.

(2) Our model also highlights that in contrast with prior work, mixed bundling actually leads to higher incentives for innovation by firms which in turn lead to enhanced quality of security software. This research investigates issues facing firms engaged in new product development and innovation in the digital economy.

(2) We have provided empirical validation of some of our analytical results using a unique data set from Amazon.com. We find support for our theory. Further, using a novel methodology based on price-cost marks ups and demand elasticities, we demonstrate the optimality of mixed bundling by firms engaging in consumer software by using information on sales ranks, prices and elasticities. Although much has been said about the cannibalization of the bundle by its components or of the components by the bundle, there has been little empirical work that measures it and in fact to the best of our knowledge, there has been no such work which has studied the software industry.

There are several ways in which this work can be extended. One possibility is to explore an extension to components with asymmetric value. Another possible extension is to develop a full blown model of a multi-product oligopolist in which each consumer has a preferred product, with a positive switching costs for buying the less desired one. Because of data limitations we are unable pursue these in our current paper. But we hope our paper paves the roadmap for future research in this area.

References

[1] Adams, W. and J. Yellen (1976). Commodity Bundling and the Burden of Monopoly. *Quarterly Journal of Economics*, 90: 475-498.

[2] Arora, A., R. Telang, and H. Xu. 2004. Optimal Policy for Software Vulnerability Disclosure. The Third Annual Workshop on Economics and Information Security (WEIS04). Minneapolis, MN, May 2004.

[3] Arora, A., J. Caulkins, R. Telang. 2006, "Sell First, Fix Later: Impact of Patching on Software Quality", *Management Science.*

[4] Bakos, Y. and E. Brynjolffson (1999). Bundling Information Goods: Pricing, Profits and Efficiency. *Management Science*, 45(12): 1613-1630.

[5] Bakos, Y. and E. Brynjolffson (2000), "Bundling and Competition on the Internet," *Marketing Science*, 19(1): 63-82

[6] Brynjolfsson, E., Y. Hu, and M. Smith (2003). Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety. *Management Science* 49(11) 1580-1596.

[7] Cavusoglu, H., H. Cavusoglu, and S. Raghunathan. 2004. How Should We Disclose Software Vulnerabilities. Workshop on Information Technology and Systems (WITS), Washington, DC, December 2004.

[8] Cavusoglu, H., B. Mishra, S. Raghunathan. 2002. Assessing the value of detective control in IT security, Proceedings of the 8th AMCIS. 1910-1918, Dallas, TX.

[9] Chen, P. and L. Hitt (2004),"Bundling with Customer Self-Selection: A Simple Approach to Bundling Low Marginal Cost Goods,' Working Paper.

[10] Chevalier, J., and A. Goolsbee (2003). Measuring Prices and Price Competition Online: Amazon and Barnes and Noble. *Quantitative Marketing and Economics* 1(2) 203-222.

[11] Chidamber, S., C. Kemerer. 1994. A metric suite for object-oriented design. *IEEE Transactions on Software Engineering* 20, 476-492.

[12] Crawford, G. (2004), "The Discriminatory Incentives to Bundle in the Cable Television Industry,' Economics Working Papers in Oxford, October.

[13] Ellison, G. and S. Ellison (2005). Search, Obfuscation and Price Elasticities on the Internet. Working Paper, MIT.

[14] Fang, H. and P. Norman (2003), "An Efficiency Rationale for Bundling of Public Goods,' Cowles Foundation Discussion Paper No. 1441; SSRI Working Paper No. 2003-19 October.

[15] Ghose, A., M. Smith and R. Telang (2006). Internet exchanges for used books: An empirical analysis of product cannibalization and welfare impact, *Information Systems Research* 17(1) 3-19.

[16] Ghose, A. and A. Sundararajan 2006. Evaluating Pricing Strategy Using eCommerce Data: Evidence and Estimation Challenges, *Statistical Science*, 21(2) May 2006.

[17] Hausman, J. (1994). Valuation of New Goods Under Perfect and Imperfect Competition. NBER working paper.

[18] Manelli, A. and D. Vincent (2005), Bundling as an Optimal Mechanism for a Multiple-Good Monopolist,' Journal of Economic Theory, Forthcoming.

[19] McAfee, P., J. McMillan, and M. Whinston (1989). Multiproduct Monopoly, Commodity Bundling, and Correlation of Values.*Quarterly Journal of Economics*, 104: 371-384.

[20] Nalebuff, B. (2004a), "Bundling as a Way to Leverage Monopoly' Yale SOM Working Paper, September.

[21] Nalebuff, B. (2004b). Bundling as an Entry Barrier. *Quarterly Journal of Economics*, 119(1): 159-187.

[22] Nambisan, S. 2003. Information Systems as a Reference Discipline for New Product Development. MIS Quarterly 27(1) 1-18

[23] Pfleegar C.P.(1997). The Fundamentals of Information Security. *IEEE Software*, 14(1), 15-17.

[24] Schmalensee, Richard (1982), "Commodity Bundling by Single-product Monopolies," *Journal of Law and Economics*, 25: 67-71.

[25] Shapiro, C. and H.R. Varian. 1999. Information Rules. Harvard Business School Press, Boston, MA

[26] Slaughter, S., D. Harter, M.S. Krishnan. 1998. Evaluating the cost of software quality. *Communications of the ACM* 41 (8), 67-73.

[27] Slaughter, S., D. Harter, M.S. Krishnan. 2000. Effects of Process Maturity on quality, cost and cycle time in software product development. *Management Science* 46, 451-466.

6 Appendix

6.1 Proofs of Propositions

Proof of Lemma 1

Let $c = c_o + c_u$.In the absence of bundling, the demand D_c for the components is given by

$$2(1-r)(1-\tfrac{p_c}{s}).$$

This implies that profits are given by

$$(p_c - c)2(1-r)(1-\tfrac{p_c}{s}).$$

Optimizing this with respect to the price, we get the optimal price =

$$p_c = \frac{s+c}{2},$$

and profits are given by

$$\pi = \frac{(1-r)(c-s)^2}{2s}. \tag{21}$$

Proof of Lemma 2

In the absence of components, when the firm introduces only the bundle, the demand D_b for the bundle is as follows:

$$(1-r)^2 + 2r(1-r)(1-\frac{p_b}{s}),$$

which leads to the following profit equation

$$\pi = (p_b - c)\left((1-r)^2 + 2r(1-r)(1-\frac{p_b}{s})\right).$$

Solving for the optimal price we get

$$\begin{aligned} p_b &= \frac{2cr + rs + s}{4r} \\ &= \frac{s(1+r)}{4r} + \frac{c}{2} \qquad (22)\\ &= \frac{s+c}{2} + \frac{s(1-r)}{4r}. \qquad (23)\end{aligned}$$

which leads to the following optimal profit equation

$$\pi = \frac{(1-r)(rs+s-2cr)^2}{8rs}. \tag{24}$$

If the seller were to price the bundle p_b at s, then its profits would be given by

$$\pi = (s-c)(1-r)^2. \tag{25}$$

Hence, the critical value of r at which the seller prices $p_b = s$, is obtained by comparing equations (24) and (25) and this value is given by:

$$r = \frac{s}{3s-2c}.$$

This implies that the pure bundle price $p_b = s$ if $r \leq \frac{s}{3s-2c}$ whereas for $r \geq \frac{s}{3s-2c}$, the pure bundle price is given by 23.[11]

Proof of Proposition 1

Consider that the seller prices the bundle at p_b and the components at p_c. First it is immediate that $p_b \leq s$ and $p_b \leq 2p_c$ since otherwise no one buys the bundle and $p_b > p_c$, since otherwise no one buys the components.

The demand from the segment of size $2r(1-r)$ is similar to that when the firm offers only pure components and is given by

$$2(1-r)(1-\frac{p_c}{s}).$$

The firm also gets demand from the segment of size $(1-r)^2$. Since $p_b \leq s$ all customers buy a product. In particular, customers buy their preferred individual components if

$$s(1-x) - p_c > s - p_b,$$

and the buy the bundle otherwise. Hence, the demand for the components from the segment of size $(1-r)^2$ is given by

$$(1-r)^2\frac{(p_b-p_c)}{s}.$$

The demand for the bundle from each segment is given by

$$(1-r)^2(\frac{1}{2} - \frac{(p_b-p_c)}{s}).$$

This implies that the total profit equation is given by

$$(p_c-c)\left(2r(1-r)(1-\frac{p_c}{s}) + 2(1-r)^2\frac{(p_b-p_c)}{s}\right) + (p_b-c)\left(2(1-r)^2(\frac{1}{2} - \frac{(p_b-p_c)}{s})\right).$$

[11] It is to verify that the optimal price given by (23) is decreasing in r.

From the first order conditions with respect to the prices, we have the following reaction functions:

$$\begin{aligned} p_b(p_c) &= \frac{4p_c + s}{4}, \\ p_c(p_b) &= \frac{r(c+s) + 2p_b(1-r)}{2}. \end{aligned}$$

Solving these equations simultaneously we get the following optimal price functions:

$$\begin{aligned} p_b &= \frac{s+c}{2} + \frac{s}{4r}, \\ p_c &= \frac{s+c}{2} + \frac{s(1-r)}{4r}. \end{aligned}$$

Equating $p_b = s$, we get the critical value of r given by $r^* = \frac{s}{2(s-c)}$. Thus, below this value of r, the seller prices the bundle at $p_b = s$. The corresponding price of the components when $r < r^*$ is given by substituting the optimal pure bundle price into the firm's profit equation and then maximizing it with respect to p_c.This yields an optimal component price of[12]

$$p_c = \frac{s+c}{2} + \frac{(s-c)(1-r)}{2}.$$

Proof of Proposition 2

The seller's optimal profit functions under mixed bundling are given as follows: When $r \geq r^*$

$$\pi = \frac{(1-r)(c^2r^2 - 2c(1-r+r^2)s + (2-2r+r^2)s^2}{2s} \tag{26}$$

When $r < r^*$,

$$\pi = \frac{(1-r)(4c^2r^2 - 4c(1+r)s + (1+3r)s^2)}{8rs} \tag{27}$$

Comparing these profit expressions with the optimal profit equations (24) and (25) when the firm offers pure bundling and with 21 when the firm offers pure components, we find that the firm's profits are always higher with mixed bundling for any value of $r > 0$.

Proof of Proposition 3

Define $\pi^M(s)$ as the gross profits at a quality level of s. We know that the level of quality that maximizes profits solves $\max \pi^M(s) - F(s)$ where

$$\pi^M(s) = \frac{(1-r)(4c^2r^2 - 4cr(1+r)s + (1+3r)s^2)}{8rs}. \tag{28}$$

Let s_M^* solve $\pi_1^M(s) = F_1(s)$. Now

[12] We verified continuity at $r = \frac{s}{2(s-c)}$.

$$\pi_1^M(s) = \frac{1}{8}\ (3 + \frac{1}{r} - \frac{4c^2 r}{s^2}),$$

and

$$\pi_1^M(s_M^*) = F_1(s_M^*).$$

Now the gross profits for the case when the firm offers only components is

$$\pi^c(s) = \frac{(1-r)(s-c)^2}{2s}.$$

We know that s_c^* solves $\pi_1^c(s) = F_1(s)$. Further,

$$\pi_1^c(s) = \frac{1}{2} - \frac{c^2}{2s^2},$$

and

$$\pi_1^M(s_c^*) = F_1(s_c^*).$$

Next, the difference in the slope of the firm's profits between the mixed bundling and pure component cases is given by

$$\pi_1^M(s) - \pi_1^c(s) = \frac{1}{8}(\frac{1}{r} - 1) + \frac{c^2}{2s^2}\ (1-r) > 0.$$

And we know that

$$\pi_{11}^M(s) = \frac{c^2 r}{s^3} < \frac{c^2}{s^3} = \pi_{11}^c(s).$$

Assume that $F_{11}(s) > \pi_{11}^c(s)$. Therefore, if there is an interior solution for both s_M^* and s_c^*, it follows that $\pi_1^M(s) > F_1(s)$ for $s < s_M^*$ and $\pi_1^M(s) < F_1(s)$ for $s > s_M^*$. Since $\pi_1^c(s_c^*) = F_1(s_c^*)$, it follows that $\pi_1^M(s_c^*) > F_1(s_c^*)$ and hence, $s_c^* < s_M^*$. [13]

[13]The proof for the comparison of the mixed bundling cases with the pure bundling cases is similar. Due to space constraints the remaining proof and the proof of other results have been omitted but are available from the authors upon request.

6.2 Methodology for Empirical Testing the Optimality of Mixed Bundling

Consider the scenario of 1 bundle with 2 components, with bundle and component prices being given by p_b, p_{c_1} and p_{c_2} respectively. For simplicity, we assume that $p_{c_1} > p_{c_2}$. We are interested in determining the optimal pure bundle price p. Let the new pure bundle price be equal to p_b. The existing profits (under mixed bundling) are given by

$$\pi_E = (p_b - c)\ Q_b + (p_{c_1} - c)\ Q_{c_1} + (p_{c_2} - c)\ Q_{c_2}. \tag{29}$$

Therefore, the demand change due to the reduction of the bundle price from P_b to P is:

$$\Delta\ Q_1 = Q_b\ \eta_{bb} \frac{(p - p_b)}{p_b}. \tag{30}$$

Similarly the the demand change due to the increase of the component prices from p_{c_1} and p_{c_2} to p is:

$$\Delta\ Q_2 = Q_b \left(\eta_{bc_1} \frac{(p - p_{c_1})}{p_{c_1}} + \eta_{bc_2} \frac{(p - p_{c_2})}{p_{c_2}} \right). \tag{31}$$

Therefore the new profits under the assertion that is there is no demand for components is:

$$\pi_N(p) = (p - c)\ Q_b \left(1 + \eta_{bb} \frac{p - p_b}{p_b} + \eta_{bc_1} \frac{(p - p_{c_1})}{p_{c_1}} + \eta_{bc_2} \frac{(p - p_{c_2})}{p_{c_2}} \right).$$

Let $K(p) = \frac{\pi_E}{\pi_N(p)}$. Hence,

$$K(p) = \frac{1}{(p - c)G(p)} \ \left((p_b - c) + (p_{c_1} - c)\ \frac{Q_{c_1}}{Q_b} + (p_{c_2} - c)\ \frac{Q_{c_2}}{Q_b} \right), \tag{32}$$

where

$$G(p) = \left(1 + \eta_{bb} \frac{p - p_b}{p_b} + \eta_{bc_1} \frac{(p - p_{c_1})}{p_{c_1}} + \eta_{bc_2} \frac{(p - p_{c_2})}{p_{c_2}} \right). \tag{33}$$

From this the value of P that minimizes K(P) is the value of P that maximizes $\Pi_N(p)$. Note that $G'(p) = \left(\frac{\eta_{bb}}{p_b} + \frac{\eta_{bc_1}}{p_{c_1}} + \frac{\eta_{bc_2}}{p_{c_2}} \right)$, which is a constant. This implies that there are 2 cases.

Case 1: $G'(P) > 0$. This implies that $(p - c)G(p)$ is maximized at $p = p_b$, assuming that we only consider the range $p \in [p_c, p_b]$. However the change in profits from choosing $p_{c_1} = p_{c_2} = p_b$ is what we compute to confirm that pure bundling at price p_b is not better than mixed bundling. Thus as long as $G'(p) > 0$, we can conclude that mixed bundling is optimal.

Case 2: $G'(p) < 0$. The F.O.C. for minimizing K(P) yields $p^* = c + \frac{G(P*)}{-G'(P^*)}$. Since $(p - c)G(p)$ is quadratic, p^* is the unique solution. Thus if $K(p^*) > 1$, then $\Pi_E > \Pi_N(p^*)$ which in turn implies that mixed bundling is optimal.