Available online at www.sciencedirect.com

ScienceDirect

Journal of Strategic Information Systems 16 (2007) 173–186

Strategic Information Systems

www.elsevier.com/locate/jsis

Preventing application software piracy: An empirical investigation of technical copy protections

Petar Djekic [1], Claudia Loebbecke *,[2]

Department of Media Management, University of Cologne, Pohligstr. 1, 50969 Cologne, Germany

Available online 12 July 2007

Abstract

To counteract application software piracy, software publishers have been implementing preventive technical copy protections into their software products. However, scientific research has not yet empirically investigated to what extent technical copy protections avoid illegal copying. Investigating this question, the paper studies the influence of technical copy protections on application software piracy. We apply descriptive statistics and a binary logistic regression to data collected from a survey of international software users. We show that technical protections fail in protecting application software from being illegally copied; none of the measures studied significantly avoids piracy. From this, we firstly derive implications for software publishers and researchers and secondly suggest directions for future research.

Keywords: IS security; Software piracy; Technical copy; Protections; Protection strategy

1. Introduction

Piracy of application software is a serious economic issue to the software publishing industry. The Business Software Alliance (BSA) speaks of an annual global loss of 13 billion US Dollar (Business Software Alliance, 2003). Piracy in this case refers to illegal

* Corresponding author. Tel.: +49 221 470 5364.
E-mail addresses: petar.djekic@uni-koeln.de (P. Djekic), claudia.loebbecke@uni-koeln.de (C. Loebbecke).

[1] Petar Djekic was a PhD student at the Department of Media Management, University of Cologne, Germany.

[2] Claudia Loebbecke holds the chair of Media Management and is Director of the Department of Business Administration and Media Management at the University of Cologne, Germany. During 2005–2006, she served as President of the Global Association for Information Systems (AIS). Her vita and publications can be found under www.mm.uni-koeln.de.

0963-8687/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jsis.2007.05.005

copying or theft of application software and associated activities such as selling illegal copies online (Straub and Collins, 1990; Traphagan and Griffith, 1998).

To counteract application software piracy, software publishers have been implementing copy protections into their software products (Conner and Rumelt, 1991; Prasad and Mahajan, 2003; Shy and Thisse, 1999). However, it seems that all technical copy protections are vulnerable (Anckaert et al., 2004; Devanbu and Stubblebine, 2000; Peace et al., 2003). For instance, merely add-on protection codes may be removed from the application software. Once this occurs, the software itself remains fully functional and can then easily be copied or shared via Internet services (Hinduja, 2003).

Nevertheless, it remains unclear, which role protections play in the battle against application software piracy. Appropriate assessments of the effectiveness of various technical copy protection measures are necessary to justify their deployments and facilitate their subsequent improvements (Benbasat and Zmud, 2003; March and Smith, 1995).

2. Research context and objective

Following Straub and Collins (1990), application software piracy refers to illegal copying and hence theft of application software. Since copying and theft may be conducted using a computer, application software piracy can be denoted as computer abuse. Measures against computer abuse have been classified into deterrent and preventive controls (Gopal and Sanders, 1997; Kankanhalli et al., 2003; Straub and Collins, 1990; Tittle, 1980).

The objective of *deterrent controls* is to influence individuals so that they avoid criminal behavior due to perceived threat or fear of sanctions resulting from the criminal activity. In case of application software piracy, such deterrent controls include guidelines or policies on legitimate use (e.g., educational, investigative, or legal campaigns) as well as security briefings or audits (Gopal and Sanders, 1997). The positive influence of deterrent controls on application software piracy has been empirically investigated and acknowledged (e.g., Gopal and Sanders, 1997; Peace et al., 2003).

Preventive controls try to actively reduce the occurrence of crime by forcing abusers to expend and deplete resources in their pursuit of criminal behavior (Kankanhalli et al., 2003). Locks in conventional doors are an example for preventive controls. Technical copy protections on software are another example; such protections hamper duplication of application software and hence impede engaging in software piracy (Gopal and Sanders, 1997).

While Anckaert et al. (2004, p. 64) notice that "none of the existing techniques for software piracy prevention provide adequate protection, since all of them have been broken", other studies (e.g., Conner and Rumelt, 1991; Givon et al., 1995; Prasad and Mahajan, 2003; Shy and Thisse, 1999) assume a positive influence.

In spite of such contradictions, we are currently not aware of any study that empirically investigates the influence of technical copy protections on application software piracy. Trying to fill this gap, this paper takes an exploratory approach and empirically investigates the influence of technical copy protections on application software piracy among end-users of application software.

To focus the scope of our investigation, we analyze technical copy protections of sequencer software products. As a particular kind of application software, sequencer software is used for creating and arranging music songs and controlling external music equipment like synthesizers. Being an industry-specific, high value niche product with low network effects, sequencer software gives us a representative case of application software that should be protected (Conner and Rumelt, 1991; Stolpe, 2000). A broad range of technical copy protections is used for the different sequencer software products studied.

3. A research framework for investigating the influence of technical protections on application software piracy

Following Anckaert et al. (2004), Gopal and Sanders (1997), and Prasad and Mahajan (2003), we distinguish between software-based and hardware-based technical copy protections. In addition, we include three personal context variables to capture possible variations in application software piracy not explained by technical copy protections (see Fig. 1).

3.1. Software-based protections

Software-based protections add specific functions to application software. They display either as software token, watermarking, obfuscation, or encryption (Albert and Moorse, 1982; Collberg et al., 2002; Devanbu and Stubblebine, 2000; Herzberg and Pinter, 1987; Peyravian et al., 2003). Watermarking, encryption, and obfuscation are embedded into

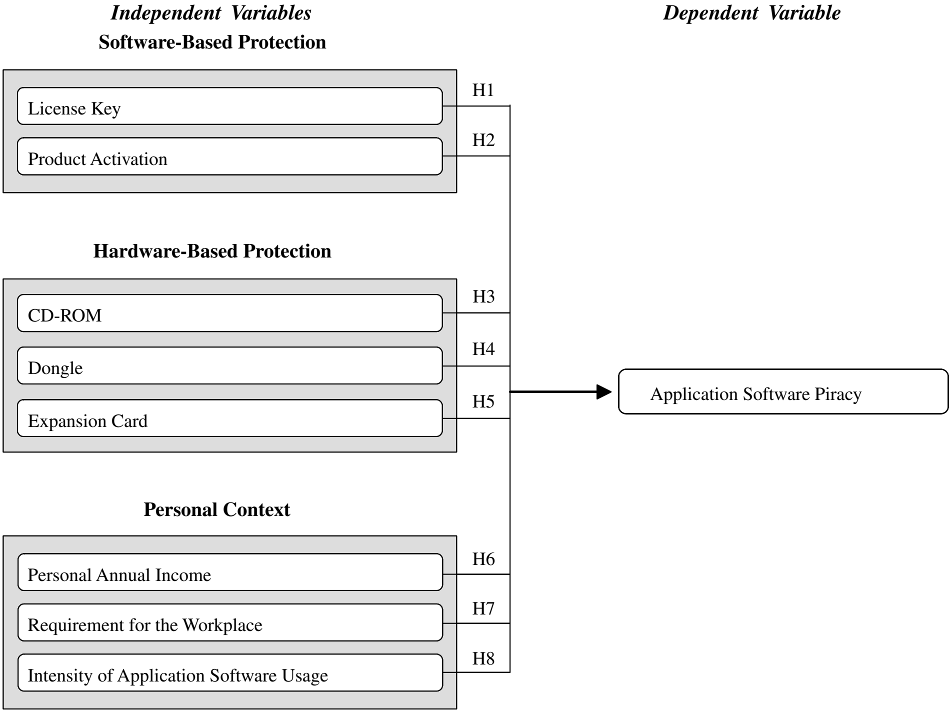

Fig. 1. Research framework for investigating the influence of technical protections on application software piracy.

the application software source code and do not interfere with users. They do not directly affect the ability to copy software and therefore do not count as preventive control. Hence they are excluded from our framework.

Software tokens describe items checked by the application software upon installation, during runtime, or at specific points in time. Lack of the software token prevents further execution of the application software (Anckaert et al., 2004; Devanbu and Stubblebine, 2000).

In this study, we analyze two different software tokens: *License Key* and *Product Activation* (Anckaert et al., 2004). The protection *License Key* employs a string composed of numbers and letters as token, which has to be entered during the installation of the application software. While the string appears to be random, highly sophisticated algorithms often underlie its generation and authentication. The *License Key* is usually provided with the application package or received by e-mail after registration at the publisher. This type of protection can be circumvented if the token is shared among individuals, as a single key can be used for installing the application software on different computer systems.

Product Activation also uses a string composed of numbers and letters as software token. With *Product Activation* the software token is unique for each software installation and depends on immutable identifiers derived from the components, such as the CPU ID or the MAC address of the Ethernet card. The *Product Activation* software token cannot be reused to install the same application software on different computer systems; a new token needs to be obtained from the software publisher for each software installation and computer system. Unlike *License Keys* or hardware-based protections, *Product Activation* makes application software 'immobile' as installation on different computer systems needs approval by the software publisher. To circumvent this type of protection, usually a modification of the application software code is necessary. Another approach are key generators or 'keygens', small applications that generate valid tokens without the need to contact the software publisher. *Product Activation*, for instance, is currently implemented in Microsoft's Windows XP operating system (Microsoft, 2006).

We propose the following two hypotheses referring to software-based protections for our research framework:

Hypothesis 1. Presence of License Key protection leads to less application software piracy.

Hypothesis 2. Presence of Activation Code protection leads to less application software piracy.

3.2. Hardware-based protections

Hardware-based protections employ hardware tokens, which are required to successfully install or run the protected application software. Hardware tokens are physical items that cannot easily be duplicated or generated. They are expected to provide stronger copy protection than software-based protection schemes using software tokens (Stolpe, 2000).

We analyze the following hardware-based protections: *CD-ROM*, *Dongle*, and *Expansion Card* (Anckaert et al., 2004; Atallah and Jiangtao, 2003; Devanbu and Stubblebine, 2000; Maude and Maude, 1984; Shi et al., 2004).

For the protection *CD-ROM*, one or more CDs are used as hardware token. Similar to protection based on floppy discs (Prasad and Mahajan, 2003), the hardware token is physically tampered to prevent its duplication. To break this protection a modification of the protected application software or duplication of the token using a CD burner is necessary. The protection *Dongle*, also referred to as hardware-key protection (e.g., Maude and Maude, 1984), uses a small stick as hardware token that has to be plugged into a USB, serial, or parallel port. The hardware token needs to be connected to the appropriate port during the software runtime. To circumvent this type of protection, the application software needs to be modified in order to spare the token checking. The alternative creation of counterfeit tokens is not used much due to the costly distribution of the hardware token along with the pirated application software. *Expansion Card* protection means a PCI-card integrated into the computer as hardware token, which is also used for core functionalities of the application software (i.e., sound processing, audio interface). This type of protection is among the strongest protections as the PCI-Card is hard to be duplicated and modification of the application software is not sufficient. Rewriting large parts of application code would be needed to replace the missing functionality of the PCI-Card.

We propose three research hypotheses regarding hardware-based protections:

Hypothesis 3. Presence of CD-ROM protection leads to less application software piracy.

Hypothesis 4. Presence of Dongle protection leads to less application software piracy.

Hypothesis 5. Presence of Expansion Card protection leads to less application software piracy.

3.3. Personal context variables

We also integrate three variables, *Personal Annual Income*, *Requirement for the Workplace*, and *Intensity of Application Software Usage*, drawn from Cheng et al. (1997) into our research framework. These three variables are included to determine the influence of sample characteristics (e.g., Hitt and Brynjolfsson, 1996; Zhu et al., 2003) and account for data variations not captured by our variables derived from software-based and hardware-based copy protections.

Personal Annual Income and price of software have often been found to determine software piracy (Cheng et al., 1997; Gopal and Sanders, 2000; Moores and Dhillon, 2000; Peace et al., 2003). Whether individuals perceive application software as too expensive, and hence not affordable, depends on their financial resources (Cheng et al., 1997).

In the work of Cheng et al. (1997) individuals cite *Requirement for the Workplace* and *Intensity of Application Software Usage* as major reasons to purchase application software.

Three additional hypotheses are proposed:

Hypothesis 6. Higher personal annual income leads to less application software piracy.

Hypothesis 7. Higher requirement of application software for the workplace leads to less application software piracy.

Hypothesis 8. Higher usage intensity of application software leads to less application software piracy.

4. Research method

4.1. Operationalization of variables

The dependent variable *Application Software Piracy* is measured as binary variable with the value '1' for a legal installation of sequencer software and the value '0' for an illegal, pirated installation (Business Software Alliance, 2003; Chiang and Assane, 2002). A pirated sequencer software installation indicates that the implemented technical copy protection measure was broken or circumvented. [Application software is protected under the 'Berne Convention for the Protection of Literary and Artistic Works' (World Intellectual Property Organization, 1979); copyright infringements can be enforced accordingly (World Intellectual Property Organization, 2006)].

To assess whether a sequencer software installation is legal or pirated, we first solicited which sequencer software was installed on the respondent's computer system. We then asked – on a separate page – for which of the previously mentioned installations a valid license was available.

The five independent software- and hardware-based protection variables are also measured as binary. The corresponding variables are *License Key* (*LK*), *Product Activation* (*PA*), *CD-ROM* (*CDROM*), *Dongle* (*DONGLE*) and *Expansion Card* (*EC*). The value '1' for a variable indicates that the specific copy protection is implemented. The '0' stands for the absence of the specific protection. Table 1 depicts the different sequencer software products and the corresponding technical copy protections as investigated in our study.

The variable *Personal Annual Income* (*INC*) is measured as an individual's annual income in US Dollar. *Requirement for the Workplace* (*WORK*) measures if an individual employs sequencer software privately or for business. It is approximated via the percentage of music-specific income over personal income. *Intensity of Application Software Usage (UINT)* indicates how much time an individual spends with the sequencer software; it is measured in hours per week.

4.2. Data collection

An online survey was conducted between November 8 and December 31, 2003. To leverage generalizability of our research results across different types of sequencer software users (i.e., to increase external validity), we collected data from professionals and amateurs as two distinct groups of sequencer software users (Straub et al., 2004).

The professional group consisted of 219 professional users of sequencer software. Professional users apply sequencer software intensely and for business purposes. Data for the professional group was gathered from an international e-mail survey among

Table 1
Sequencer software and corresponding technical copy protections

Protection	Sequencer software
License key	Cakewalk sonar, synapse orion
Product activation	Imageline fruityloops, mackie tracktion
CD-ROM	MOTU digital performer, propellerheads reason
Dongle	Apple logic, steinberg cubase, steinberg nuendo
Expansion card	AVID protools

2742 musicians ($n = 2742$). The required e-mail addresses were collected from three musicians' Internet yellow pages. A cover letter including an individual key which granted access to the survey website was sent to all e-mail contacts ($n = 2742$). Of the 2742 invitations sent, 219 resulted in a completely filled-out questionnaire (8%). E-mail feedback indicated two reasons for the low response rate: The cover letter was regarded as spam and e-mail addresses were outdated. Since the key provided with the cover letter was unique for each participant, nobody could fill out multiple questionnaires.

Data from the amateur group was collected by surveying members of thirteen message boards for music software. The size of the message board population was not known in advance. To calculate the response rate, we estimated a total population size of 2159 individuals based on the number of clicks on the survey invitation site 575 completely filled-out questionnaires led to a 27% response rate for the amateur group. Multiple questionnaires per person were avoided by placing a cookie with the survey status and by saving the IP-address of each participant.

To assure the difference of both data sets, we tested for equal means of the variables *Intensity of Application Software Usage* and *Requirement for the Workplace*. As we found a significant difference (p-value < 0.05) for both values, each group data was analyzed separately.

We assessed the non-response bias by comparing the gender and age between early and late respondents as well as by comparing the percentage of male readers in each group to a reference value.

The method of comparing answers from early and late respondents is based on the extrapolation procedure suggested by Armstrong and Overton (1977). It has been used successfully in different studies for testing non-response bias (e.g., Ba and Pavlou, 2002; Heide and Weiss, 1995). Our null hypothesis of equal means across early and late respondents was not rejected. The t-test's p-values were above 0.05 for gender and age in the professional and amateur group.

Testing for non-response bias by comparing data characteristics to a reference value is a known method in IS research and used for example by Ba and Pavlou (2002). In our study, we compared the percentage of male readers in the professional and amateur group (92% and 98% respectively) to the percentage of male subscribers to the journal 'Electronic Musician' (Electronic Musician, 2005). The t-test's p-value was above 0.05 for professional and amateur group, indicating no significant difference between values. Based on the two tests, non-response bias did not appear to be a serious issue.

4.3. Descriptive statistics

We first descriptively analyzed our data by calculating regional software piracy rates. We measured illegal copying in six geographic regions and also calculated application software piracy rates for each protection measure.

Pirated software installations were those without license. The Piracy Rate (PR) was the percentage of pirated application software installations, e.g., pirated sequencer software installations, over all sequencer software installations:

$$\mathrm{PR} = \frac{\text{Number of pirated sequencer installations}}{\text{Total number of sequencer installations}}$$

For six regions, we compared piracy rates from our survey to the rates published by the BSA for standard application software like spreadsheets or word processors. One would expect the piracy rates of the copy-protected sequencer software in our study to be below the rates of the unprotected standard application software in the BSA figures.

4.4. Binary logistic regression

As our dependent variable was dichotomous, we tested our research framework and the corresponding hypotheses using a binary logistic regression. In the IS domain, logistic regression models had for example been used by Chau and Tam (1997) or Zhu et al. (2003).

Following our research framework presented in Fig. 1, we specified the following two logit models, one for software-based protections and one for hardware-based protections:

$$\ln[\mathrm{SI}/1-\mathrm{SI}]_{\mathrm{Software}} = \alpha + \beta_1 \cdot LK + \beta_2 \cdot PA + \beta_3 \cdot INC + \beta_4 \cdot WORK + \beta_5 \cdot UINT$$

$$\ln[\mathrm{SI}/1-\mathrm{SI}]_{\mathrm{Hardware}} = \alpha + \beta_1 \cdot CDROM + \beta_2 \cdot DONGLE + \beta_3 \cdot EC + \beta_4 \cdot INC + \beta_5 \cdot WORK + \beta_6 \cdot UINT$$

Testing our hypotheses meant testing for statistically significant, positive coefficients. Significant, positive coefficients raised the probability of a sequencer software installation being legal and hence positively influenced application software piracy (lower piracy rates). Negative and significant coefficients lowered the probability and indicated a negative influence on application software piracy, i.e., higher piracy rates.

The overall model fit of our binary logistic regression was assessed using three Goodness-of-Fit tests. The *Likelihood Ratio Test* (*LR*) analyzes if the independent variables have explanatory power. Its significance indicates a good fit of the model to the data (Menard, 1995). The interpretation of the R^2-*Nagelkerke* value, the second Goodness-of-Fit test, is similar to the R^2 in a linear regression. With more than 20% of the dependent variable's variance explained by the model, values above 0.2 indicate a good fit of the model to the data (Nagelkerke, 1991). The *Hosmer–Lemeshow* (*HL*) Chi, our third Goodness-of-Fit test, compares one's own model to a model with a perfect fit. A non-significance of the test indicates a good fit of the model to the data (Hosmer and Lemeshow, 2000).

5. Research results

5.1. Descriptive statistics

Table 2 compares the regional software piracy rates from our survey to the rates from the BSA study (Business Software Alliance, 2003). The world piracy rate shows the piracy rate over all regions. Piracy rates for each technical copy protection are displayed in Table 3.

Table 2
Regional software piracy rates (in %)

Region	Professional group	Amateur group	BSA
World piracy rate	19	22	39
North-America	10	17	24
Western Europe	24	25	35
Middle east/Africa	No data	27	49
Latin America	No data	67	55
Asia/Pacific	21	19	55
Eastern Europe	100	63	71

Table 3
Application software piracy rates per technical copy protection (in %)

Technical copy protection	Professional group	Amateur group
Software-based		
License key	25	16
Product activation	50	23
Hardware-based		
CD-ROM	24	30
Dongle	22	20
Expansion card	7	9

In our study four out of five (80%) regional piracy rates in the professional group and six out of seven (86%) in the amateur group were lower than in the BSA figures (see Table 2). In brief, 'our' protected sequencer software was less pirated than the unprotected software studied by the BSA. This suggested preliminary support for our expectation that technical copy protections lead to less software piracy. We also found partial support for Hypothesis 6: Higher income regions (e.g., North America) showed lower regional piracy rates than lower income regions (e.g., Eastern Europe).

However, Tables 2 and 3 also illustrate that no technical copy protection completely prevented illegal copying and hence enforced a piracy rate of zero. The protection measure *Expansion Card* achieved the lowest overall piracy rates in both groups (Table 3).

5.2. Binary logistic regression

The outcomes of the logit models for both groups and for software- and hardware-based protections are shown in Tables 4 and 5.

None of the variables related to protection had significantly positive coefficients (see Table 4). Therefore, we had to reject all our hypotheses related to protections (i.e., Hypothesis 1–5). On the other hand, all variables related to the personal context had positive coefficients and were statistically significant in both groups.

The three Goodness-of-Fit tests showed mixed results regarding the overall model fit. The *Likelihood Ratio Test*s were significant and the R^2-*Nagelkerke* value was above 0.2.

Table 4
Logit model – Software-based protections and personal context variables

Variables	Professional group		Amateur group	
	Coefficient	SE	Coefficient	SE
Software-based Protections				
License key	0.314	0.647	0.605	0.369
Product activation	−1.818	1.445	0.584	0.309
Personal context				
Personal income	0.538^{***}	0.115	0.626^{***}	0.066
Requirement f. t. workplace	0.234^{*}	0.114	0.358^{***}	0.078
Intensity of appl. softw. usage	0.229^{**}	0.066	0.044^{***}	0.009

Goodness-of-Fit; LR: 277.948^{***}; HL Chi: 19.183^{**}; R^2-N: 0.226; LR: 746.046^{***}; HL Chi: 23.306^{**}; R^2-N: 0.269; $^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$.

Table 5
Logit Model – Hardware-based protections and personal context variables

Variables	Professional Group		Amateur group	
	Coefficient	SE	Coefficient	SE
Hardware-based protections				
CD-ROM	−0.585	0.623	-0.930^{***}	0.275
Dongle	0.177	0.621	−0.401	0.270
Expansion card	0.812	0.729	0.111	0.810
Personal context				
Personal income	0.520^{***}	0.115	0.636^{***}	0.067
Requirement f. t. workplace	0.256^{*}	0.115	0.350^{***}	0.079
Intensity of appl. softw. usage	0.206^{**}	0.069	0.040^{***}	0.009

Goodness-of-Fit; LR: 269.454^{***}; HL Chi: 4.593; R^2-N: 0.260; LR: 736.170^{***}; HL Chi: 13.355; R^2-N: 0.284; $^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$.

The HL Chis, however, were significant in both groups. This indicated that the variables might have been ineligible of reliably predicting whether the installed sequencer software was pirated or legal.

The logistic regression results for hardware-based protections were similar to those for software-based protections portrayed in Table 4. The coefficients related to protection were not statistically significant. The results for the variables related to the personal context were also comparable. Coefficients were positive and statistically significant. The three Goodness-of-Fit tests promised good overall model fit in both groups: The *Likelihood Ratio Test*s were significant, the R^2-*Nagelkerke* was above 0.2, and the HL Chis were non-significant. Hence, implementations of hardware-based protections seemed to better than those of software-based protections predict whether installed sequencer software was pirated or legal.

However, such positive influence could not be directly attributed to any of the protection schemes studied, as none of the protection-related variables was significant. Hence, this result somehow pointed to a very weak link between hardware-based protections and the occurrence of sequencer software piracy.

5.3. Major findings

While our descriptive analysis indicated support for our protection-related Hypotheses 1–5, the hypotheses were not confirmed by our logistic regression analysis applied for hypothesis testing.

We found none of the software- or hardware-based protection variables to be positive and statistically significant, and hence Hypothesis 1–5 were rejected. However, the better model fit of our hardware-based logit model in terms of the HL Chi indicated an overall weak positive influence of hardware-based protections on software piracy. Overall, our analysis supports the theoretical findings by Anckaert et al. (2004) that none of the current technical copy protections can fully prevent software piracy.

The hypotheses concerning our remaining three independent variables (i.e., Hypotheses 6–8) were fully supported. Higher *Personal Annual Income*, stronger *Requirement for the Workplace*, and stronger *Intensity of Application Software Usage* increased the probability of sequencer software installations being legal.

The findings regarding the personal context variables are in line with previous research (Cheng et al., 1997; Gopal and Sanders, 2000). The variable *Personal Annual Income* appears to have the strongest influence on software piracy, as it is highly statistically significant in both groups and logit models (p-value < 0.001). Further, *Intensity of Application Software Usage* has a slightly higher statistical significance than *Requirement for the Workplace* and hence seems to have a stronger influence on software piracy.

6. Research implications and discussion

Our research has implications for research and practice: Different from the literature (e.g., Conner and Rumelt, 1991; Prasad and Mahajan, 2003; Shy and Thisse, 1999), our findings suggest hardly any influence of technical copy protection measures on sequencer software piracy – and probably the same for similar industry-specific application software. A software publisher's aim to deploy technical copy protections in order to turn former pirates into buyers and thus to positively influence software sales may not be achievable. However, similar weaknesses of protection systems for audio CDs did not hamper their deployment (Halderman, 2003).

The findings regarding personal income, professional use, and usage intensity are only of confirmatory nature. However, the impact of personal income along with comparably high piracy rates in low income regions (see Table 2) emphasizes the importance of adequate pricing strategies, which may include price differentiation by product features or by region (Gopal and Sanders, 2000; Moores and Dhillon, 2000). For example, Microsoft is shipping a cut down version of its operating system software Windows XP for USD 35 to selected countries with high software piracy (Microsoft, 2004).

Overall, we cannot recommend implementing any of the technical copy protections studied as preventive control against software piracy. These results are surprising since intuitively harder to replicate and more expensive forms of copy protection (i.e., *Dongle* and *Expansion Card*) would be expected to protect sufficiently.

However, the low piracy rates resulting from using an *Expansion Card* outline possible improvements to current protection measures. Hardware-based technical copy protections integrated into computer systems and providing core functions to the application software seem to provide better protection from software piracy and should therefore be pursued in research labs. Not token replicability, but rather token integration into the application software code is crucial in technical copy protection. Without functionalities beyond copy protection, the protection code can be successfully removed from the application software. First efforts toward token integration are taken by the Trusted Computing Platform Alliance (Arbaugh, 2002).

Overall software publishers may rather pursue a protection strategy focusing on deterrent controls like educational campaigns than on employing preventive controls such as technical copy protections (Gopal and Sanders, 1997). Focusing on deterrent controls should allow software publishers to charge higher prices for their products, because application software without technical copy protection has a higher user valuation (Shy and Thisse, 1999). In addition, by abandoning hardware-based copy-protection techniques, software publishers would be able to sell and deliver their products online, thereby reaching more potential customers than via traditional distribution channels.

7. Research limitations

Limitations of our study arise from three causes:

(1) The focus on sequencer software may aggravate generalizations for different types of software. However, similar results across professional and amateur groups indicate a high external validity of our results for different types of sequencer software users. We expect, but cannot confirm our findings to be also applicable to similar industry-specific software (e.g., graphics application software).
(2) The characteristics of online surveys, especially the anonymity of participants, complicate the verification of the face validity. The potential conflict of self-incrimination may have led to some underreporting. The quality of the data may have slightly suffered from underreporting. However, underreporting would not affect the direction of our results.
(3) The comparison to BSA data needs to remain superficial in some points, as the BSA does not document the examined software in detail in their annual piracy study (Business Software Alliance, 2003).

These limitations are important as the results of a logistic regression model depend on the characteristics of the data. A few outliers in the data (Menard, 1995) or unbalanced data (Cramer, 1999) can lead to a bad model fit. The rather weak model fit indicated by the HL Chi may be a result of aforementioned data problems. However, as both other Goodness-of-Fit tests provide satisfying results, the HL Chi may also result from lacking additional explanatory variables rather than data characteristics.

8. Summary and future research

Technical copy protections currently used to protect application software seem to play only a minor role in software piracy prevention. Personal context variables such as annual income or software usage intensity determine piracy rates more than the technical copy protection employed.

Studying software piracy only with a technical security perspective can therefore not be sufficient. Further research on social aspects and patterns of human interaction in using application software is needed (see also Dhillon and Backhouse, 2001). We suggest to reconsider the possible strategic contribution of technical copy protection usage as the introduction of technical copy protection obviously is not sufficient for software publishers to fully differentiate between paying and non-paying users. We also propose to reach beyond preventive controls and dig deeper into deterrent and economic measures such as legal campaigns, educational campaigns and regional price differentiation. The role that software usage intensity plays for software piracy demands further investigation. As legal use is closely related to acceptance, we suggest applying the Technology Acceptance Model (Davis, 1989) in such an endeavor.

References

Albert, D., Moorse, S., 1982. Combating software piracy by encryption and key management. IEEE Computer 17 (4), 68–73.

Anckaert, B., De Sutter, B., De Bosschere, K., 2004. Software piracy prevention through diversity. Proceedings of the 4th ACM Workshop on Digital Rights Management, Washington, DC, USA, pp. 63–71.

Arbaugh, B., 2002. Improving the TCPA specification. Computer 25 (8), 77–79.

Armstrong, J., Overton, T., 1977. Estimating non-response bias in mail surveys. Journal of Marketing Research 14 (3), 396–402.

Atallah, L., Jiangtao, L., 2003. Enhanced smart-card based license management. IEEE International Conference on E-Commerce, June 2003, pp. 111–119.

Ba, S., Pavlou, P., 2002. Evidence of the effect of trust building technology in electronic markets: price premiums, buyer behavior. MIS Quarterly 26 (3), 243–268.

Benbasat, I., Zmud, R., 2003. The identity crisis within the IS discipline: defining and communicating the discipline's core properties. MIS Quarterly 27 (2), 183–194.

Business Software Alliance 2003. Annual Business Software Alliance global software piracy study. <www.bsa.org/globalstudy/2003_GSPS.pdf>, access on 2006-01-01.

Chau, P., Tam, K.Y., 1997. Factors affecting the adoption of open systems: an exploratory study. MIS Quarterly 21 (1), 1–24.

Cheng, K., Sims, R., Teegen, H., 1997. To purchase or to pirate software: an empirical study. Journal of Management Information Systems 13 (4), 49–60.

Chiang, E., Assane, D., 2002. Software copyright infringement among college students. Applied Economics 34 (2), 157–166.

Collberg, C.S., Thomborson, C., Low, D., 2002. Watermarking, tamper-proofing, and obfuscation – tools for software protection. IEEE Transactions on Software Engineering 28 (8), 735–746.

Conner, K., Rumelt, R., 1991. Software piracy: an analysis of protection strategies. Management Science 37 (2), 125–139.

Cramer, S., 1999. Predictive performance of the binary logit model in unbalanced samples. The Statistican 48 (1), 85–94.

Davis, F., 1989. Perceived usefulness, perceived ease of use and user acceptance of information technology. MIS Quarterly 47 (2), 141–147.

Devanbu, P., Stubblebine, S., 2000. Software engineering for security: a roadmap. Proceedings of the International Conference on Software Engineering, Limerick, Ireland, pp. 227–239.

Dhillon, G., Backhouse, J., 2001. Current direction in IS security research: towards socio-organizational perspective. Information Systems Journal 11 (2), 127–153.

Electronic Musician 2005. Subscriber profile. <www.advertisers.emusician.com/market/>, 2005-12-27.

Givon, M., Mahajan, V., Muller, E., 1995. Software piracy – estimation of lost sales and the impact on software diffusion. Journal of Marketing 59 (1), 29–37.

Gopal, R., Sanders, G., 1997. Preventive, deterrent controls for software piracy. Journal of Management Information Systems 13 (4), 29–48.

Gopal, R., Sanders, G., 2000. You can't get blood out of a turnip. Communications of the ACM 43 (9), 83–89.

Halderman, J., 2003. Evaluating new copy – prevention techniques for audio CDs. Lecture Notes in Computer Science 2696 (2003), 101–117.

Heide, J., Weiss, A., 1995. Vendor consideration and switching behavior for buyers in high-technology markets. Journal of Marketing 59 (3), 20–43.

Herzberg, A., Pinter, S.S., 1987. Public protection of software. ACM Transactions on Computer Systems 5 (4), 371–393.

Hitt, L., Brynjolfsson, E., 1996. Productivity, business profitability, and consumer surplus: three different measures of information technology value. MIS Quarterly 20 (2), 121–142.

Hinduja, S., 2003. Trends and patterns among online pirates. Ethics and Information Technology 5 (1), 49–61.

Hosmer, W., Lemeshow, S., 2000. Applied Logistic Regression. Wiley & Sons, New York USA.

Kankanhalli, A., Teo, H.-H., Tan, B., Wei, K.-K., 2003. An integrative study of information systems security effectiveness. International Journal of Information Management 23 (2), 139–154.

March, S., Smith, G., 1995. Design and natural science research on information technology. Decision Support Systems 15 (4), 251–266.

Maude, T., Maude, D., 1984. Hardware protection against software piracy. Communications of the ACM 27 (9), 950–959.

Menard, S., 1995. Applied Logistic Analysis Quantitative, Applications in the Social Sciences. Sage, Thousand Oaks USA.

Microsoft 2006. Microsoft product activation. <www.microsoft.com/piracy/activation.mspx>, access on 2006-06-30.

Microsoft 2004. Windows XP starter edition pilot expands to Russia. <www.microsoft.com/presspass/features/2004/sep04/09-27StarterEd.asp, access on 2006-06-30.

Moores, T., Dhillon, G., 2000. Software piracy: a view from Hong Kong. Communications of the ACM 43 (12), 88–93.

Nagelkerke, D., 1991. A Note on a general definition of the coefficient of determination. Biometrika 78 (3), 691–693.

Peace, G., Galletta, D., Thong, L., 2003. Software piracy in the workplace: a model and empirical test. Journal of Management Information Systems 20 (1), 153–177.

Peyravian, M., Roginsky, A., Zunic, N., 2003. Methods for preventing unauthorized software distribution. Computers and Security 22 (4), 316–321.

Prasad, A., Mahajan, V., 2003. How many pirates should a software firm tolerate? An analysis of piracy protection on the diffusion of software. International Journal of Research in Marketing 20 (4), 337–353.

Shi, W., Lee, H., Lu, C., Zhang, T., 2004. Attacks, risk analysis for hardware supported software copy protection systems. Proceedings of the 4th ACM Workshop on Digital Rights Management, October 25, Washington, DC, USA, pp. 42–55.

Shy, O., Thisse, J., 1999. A strategic approach to software protection. Journal of Economics Management Strategy 8 (2), 163–190.

Stolpe, M., 2000. Protection against software piracy: a study of technology adoption for the enforcement of intellectual property rights. Economics of Innovation and New Technology 9 (1), 25–52.

Straub, D., Collins, R., 1990. Key information liability issues facing managers: Software piracy, proprietary databases and individual rights to privacy. MIS Quarterly 14 (2), 143–156.

Straub, D., Boudreau, M., Gefen, D., 2004. Validation guidelines for IS positivist research. Communication of the Association of Information Systems 13, 380–427.

Tittle, C., 1980. Sanctions and Social Deviance: The Questions of Deterrence. Praeger, New York.

Traphagan, M., Griffith, A., 1998. Software piracy and global competitiveness: report on global software piracy. International Review of Law, Computers & Technology 8 (2), 431–451.

Word Intellectual Property Organization 1979. Berne Convention for the Protection of Literary and Artistic Works, Paris Act of July 24, 1971, as amended on September 28, 1979. <www.wipo.int/clea/docs/en/wo/wo001en.htm>, access on 2006-06-30.

World Intellectual Property Organization 2006. Berne Convention - Contracting Parties. <www.wipo.int/treaties/en/documents/pdf/berne.pdf>, access on 2006-06-30.

Zhu, K., Kraemer, K., Xu, S., 2003. E-Business adoption by European firms: a cross-country assessment of the facilitators and inhibitors. European Journal of Information Systems 12 (4), 251–268.