EVALUATION OF ACTIVATION BASED SOFTWARE LICENSE ENFORCEMENT

by

Oleg Afonin
B.Sc. Krasnoyarsk State University, 1998
M.Sc. Krasnoyarsk State University, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER SCIENCE

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
September 2002

Department of Computer Science

The University of British Columbia
Vancouver, Canada

Date 10 Oct 2002

Abstract

Software piracy, software licensing and license control are all important issues to software developers. Small software developers did not have access to a system available to control a number of installations of licensed software products. A company could obtain a single license and use in among the network, or there could be a breach of security allowing multiple users to use fully licensed software on many different computers.

Tying software installation to hardware is the most common way to prevent or restrict those illegal activities. Hardware tying, however, has a drawback to software publishers: a user has to pass a Hardware ID to the software manufacturer at the time of ordering, which might affect his willingness to purchase that product at all.

With the development of the Internet it became important to control software licensing and distribution online to protect revenues from losses incurred by both *intended* and *casual* software piracy. An online license management system helps reduce both forms of piracy by ensuring that each copy of the software product being installed is legal and has been installed on a PC in compliance with its license terms. Installations beyond those allowed in the license agreement will fail to activate, thus preventing both casual and intended piracy.

The main question of this research is whether or not online software license activation is a effective solution of reducing software piracy. It was interesting to figure out if its direct or indirect benefits prevail over its disadvantages.

This research presents an in-depth evaluation of ActivateSoft.NET, a software activation system. It describes software activation policies as sets of parameters that allow precise control on how software end-user license agreements are enforced.

Finally, an experiment is designed and run, and raw data is collected on the usage of real-world software. Collected data is analyzed to find out how online license management can affect software usage patterns, providing a hint to answering the key question: whether or not the use of software activation reduces software piracy.

Table of Contents

List of Tables

List of Figures

1. Introduction

Software piracy, software licensing and license control are all important issues to software developers. People who use illegal software not only hurt themselves, they also contribute to a problem that cumulatively can hurt job creation locally and regionally in the software industry and related businesses. Software piracy also has a significant impact on the high-tech industry, resulting in lost jobs, decreased innovation and higher costs to consumers.

At the time of this research there was no system commercially available to independent software developers that would control number of installations of licensed software products. A company could obtain a single license and use it among the network. Users would break terms of their license agreement by using software on multiple computers without purchasing additional licenses.

There are certain ways to prevent or restrict those illegal activities. One way is tying software installation to hardware. Hardware tying, however, has a huge drawback to software publishers: a user has to pass a Hardware ID to the software manufacturer at the time of ordering, which might his willingness to purchase that product at all, playing its role in the decision of getting this or competitor's product. Additionally, a user will have to manually obtain a Hardware ID and pass it to the ordering system, which is usually implemented as a SSL-encrypted Web order form.

Product Activation is an anti-piracy method to verify a software license and limit the spread of software piracy. Product Activation ensures that the users install each software license on as many computers as it is permitted by a software license agreement. Every time a product is installed on a new PC or if there is a change in hardware of the PC on which it was installed, the user has to activate his copy of the software product by obtaining an Activation Code from the activation server. If a product is not activated within 14 days, it will cease working, and only the activation functionality will be available.

More companies are developing software activation systems. Searching for "software activation" on Google.com did not provide any meaningful links in May 2002; in September 2002 the same search returned around 20 companies requiring their users to activate a copy of a product they've just purchased.

The key question of this thesis is whether or not license enforcement with online software activation reduces software piracy by enforcing the number of PCs on which a software license can be installed. In order to answer this question I am going to try to evaluate how many users violate software license agreement by installing their software licenses on multiple PCs. I met several issues doing this research. My sample is fairly small due to the time and resource constraints, and it is hard to explain several patterns in the users' behaviour.

To answer the key question, I developed a software activation system ActivateSoft.NET. The system was used to log customers' purchasing activity and control the number of installations. I summarized and analyzed these logs, answered intermediate questions, and suggest the answer to the key question.

2. Background

This chapter presents a background of software activation, including a concept and overview of Microsoft Product Activation.

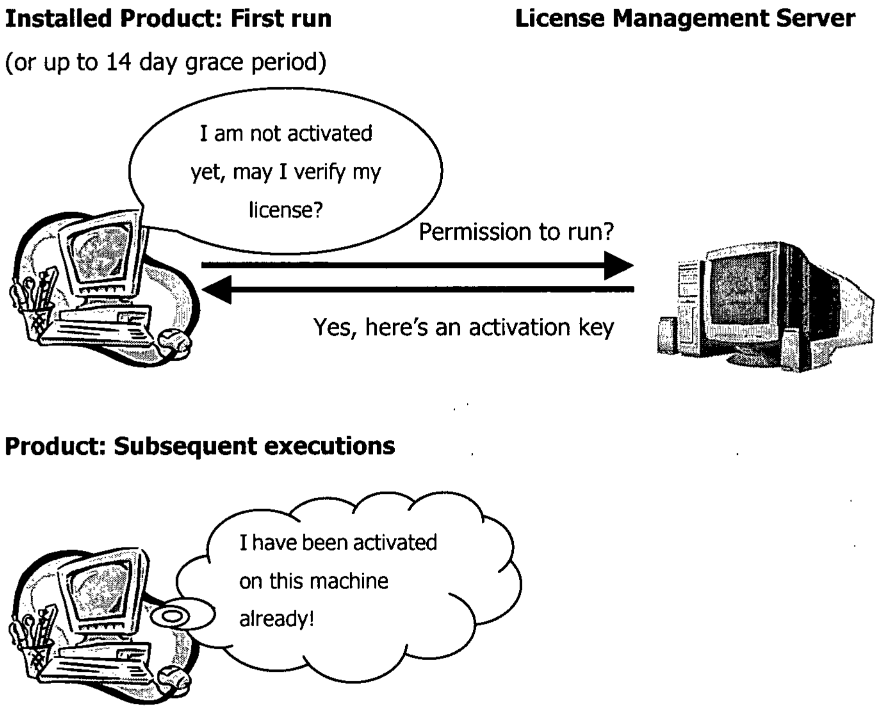

Installed Product: First run
(or up to 14 day grace period)
License Management Server
I am not activated yet, may I verify my license?
Permission to run?
Yes, here's an activation key
Product: Subsequent executions
I have been activated on this machine already!

Figure 1 - ***A Concept of Product Activation***

Microsoft was not the first company that implemented activation-based license enforcement. It was, however, the first company that did it for well-known products, Office XP and Windows XP. They have performed user studies and created an extensive list of answers to the most frequent questions their users had about the product activation. Their studies helped me a lot to develop the activation system.

Microsoft defines Product Activation as an *"anti-piracy technology designed to verify that the product has been legitimately licensed"*.

Microsoft Product Activation works by validating that the software product key, required as part of product installation, has not been used on more PCs than is allowed by the software license.

Product key information is sent along with a “hardware hash” (a non-unique number generated from the PC's hardware configuration) to Microsoft's activation system during activation.

Activation is completed either directly via the Internet or by a telephone call to a customer service representative. Microsoft allows unlimited number of activations on a single PC with the same product key. Product Activation discourages piracy by limiting the number of times a product key can be activated on different PCs.

Microsoft Product Activation is designed to be “transparent” and unobtrusive for customers who legitimately acquire a software license. Customers are able to delay activation for several uses of the product or a certain period of time, until a time that is convenient for them. For those who obtain a copy of the software illegally, Product Activation makes their inappropriate use of the product more difficult.

Microsoft fights piracy aggressively, increasingly strengthening license enforcement [15]. In early days of WPA, which stands for Windows Product Activation, Microsoft did not send Windows product keys to their activation servers during the activation (this was done to reduce possible privacy concerns). However, in Windows XP Service Pack 1 they changed their mind and now include a complete product key in the Installation ID that is necessary to activate a copy of Windows XP. Our system was finished in April 2002, almost half a year before SP1 arrived, yet we implemented the same approach, and use a Product Key as a part of our Installation ID.

RealNetworks (*www.real.com*) is another big company that requires activation of its product, RealOne Player. Both free and paid versions of the latest RealOne Player must be activated over the Internet after 3 uses, otherwise it will stop working except showing the activation dialog.

Besides Microsoft and Real there are many small companies that require their users to activate software licenses. Examples include: Bourse Data (*www.boursedata.com.au*), Kidasa Software (*www.kidasa.com*), Ericom Software (*www.ericom.com*), DataCube (*www.datacube.com*), Trading Solutions (*www.tradingsolutions.net*), MindSoft (*www.mindsoftweb.com*), and many others that can be found by performing a simple search on Google.com for “activate product”.

3. ActivateSoft.NET

This chapter describes a theory of software activation, and explains activation technology. It presents the activation system software developed to collect data and perform the study.

3.1. Activation System – The Software

3.1.1. Basics of Activation

I propose a solution similar to Microsoft Product Activation in its concept: a software activation system. A system like that was implemented by Microsoft in its Office XP and Windows XP. Our system is similar to Microsoft Windows Product Activation (WPA) in that it uses the same Product Key format, and has a license management center implemented as an online server.

Our implementation differs from the Microsoft one in that we designed ours to support multiple developers and multiple products. Currently Microsoft WPA supports Microsoft products only, as they never opened their system for third parties; they do the opposite thing by keeping most technical details a secret. We allow developers to control the strength of enforcement for each of their products. Our system is less technically advanced and less secure than Microsoft WPA, but it is open to other software developers, expandable and customizable.

Software activation happens in two steps. First, a user purchases and installs a product. A valid Product Key is required for the installation. To use the product after the installation the user will have to activate it by obtaining an Activation Code from the server. An Activation Code is a cryptographically strong digital signature of a combination of a Product Key and Hardware ID. The activation process is implemented by acquiring a digital certificate from a centralized license manager (a server connected to the Internet).

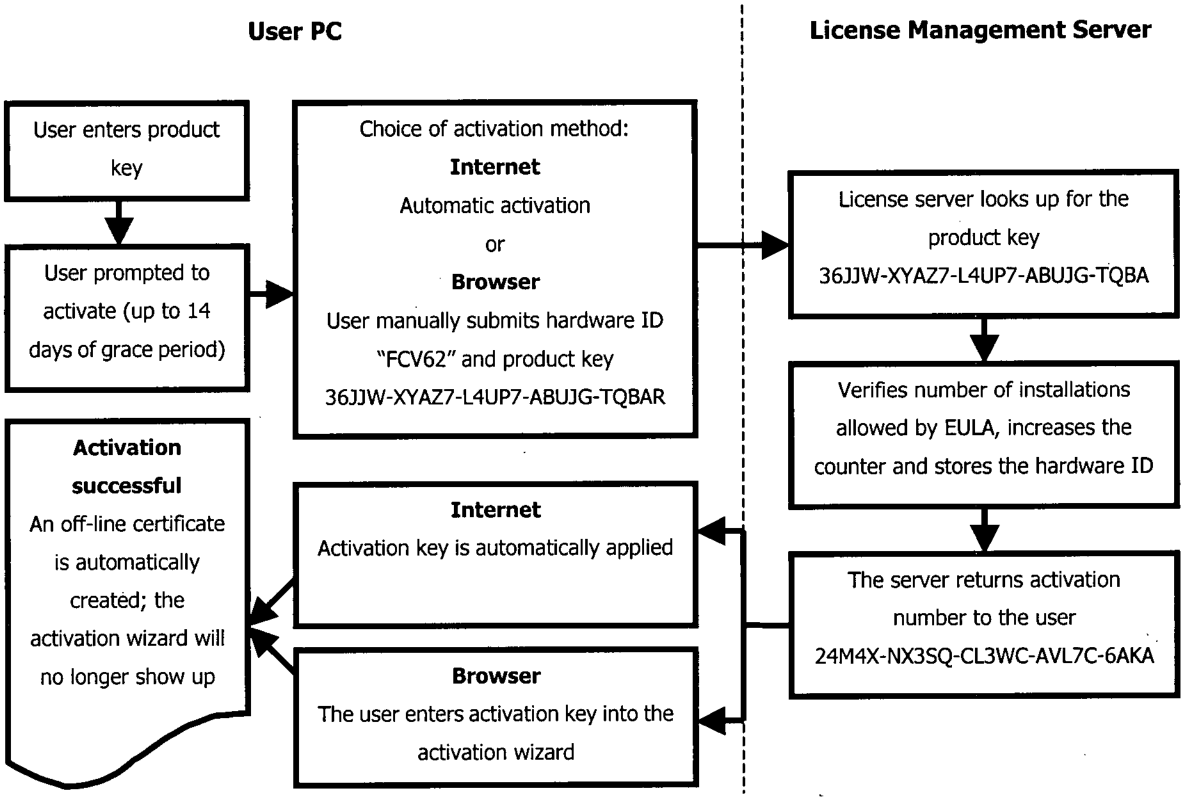

User PC
License Management Server
User enters product key
User prompted to activate (up to 14 days of grace period)
Choice of activation method:
Internet
Automatic activation
or
Browser
User manually submits hardware ID "FCV62" and product key 36JJW-XYAZ7-L4UP7-ABUJG-TQBAR
License server looks up for the product key 36JJW-XYAZ7-L4UP7-ABUJG-TQBA
Verifies number of installations allowed by EULA, increases the counter and stores the hardware ID
The server returns activation number to the user 24M4X-NX3SQ-CL3WC-AVL7C-6AKA
Internet
Activation key is automatically applied
Browser
The user enters activation key into the activation wizard
Activation successful
An off-line certificate is automatically created; the activation wizard will no longer show up

Figure 2 – Activation Process

Product activation will help reduce casual copying by ensuring that the copy of the software product being installed is legal and has been installed on a PC in compliance with the license terms. Installations beyond those allowed in the license agreement will fail to activate. Use of the activation system provides developers with invaluable statistics on their customers' software usage patterns. Each transaction is recorded and stored in the database, available for later research.

The system consists of the following parts:

1. Cryptography software (Product Key and Activation Code generation and verification)
2. SQL database (stores issued Product Keys, Activation Codes, and user activity logs)
3. Online interfaces for registering and activating software products, and Developer's Control Panel
4. C++ SDK included to ActivateSoft.NET enabled products.

3.1.2. Cryptography Software

It is very important to have strong Product Keys and Activation Codes that could not be broken. One of the worst problems experienced by software developers is a possibility of appearance of so called 'key generators', or 'keygens'. Key generators are created by hackers or hacker teams, and produce fake software licenses that are positively validated by the products they target.

For weak key generation schemes the hackers use reverse engineering in order to figure out an algorithm of validating a software license. After that they reverse the algorithm, and obtain the ability to produce fake license keys.

In order to counter this type of attack, we used a strong open key cryptography for license key generation. We implemented strong open key crypto software based on the HFE algorithm that uses a *private* key to generate license keys, and a *public* key to verify them. This is the opposite of the classical open crypto, with private keys used for decryption and verification, and public keys for encryption and signing. Our approach guarantees [Appendix 2] that it is impossible to reverse the license key verification algorithm to make a key generator, or produce a valid license code without knowing a *private key.*

Our Product Keys are generated as cryptographic signatures of a customer's name (all characters are converted to capitals, spaces and punctuation removed). Such a signature is a one-way function of a string, also known as a 'hash function'. A *private key* is required to generate a signature, and a *public key* is required to verify it. We only generate Product Keys on the ActivateSoft.NET server. A Product Key can be verified from the product executable file by using a non-secret *public key.*

Activation Codes are generated in a similar fashion, except that they sign a string formed as a concatenation of a Product Key and Hardware ID.

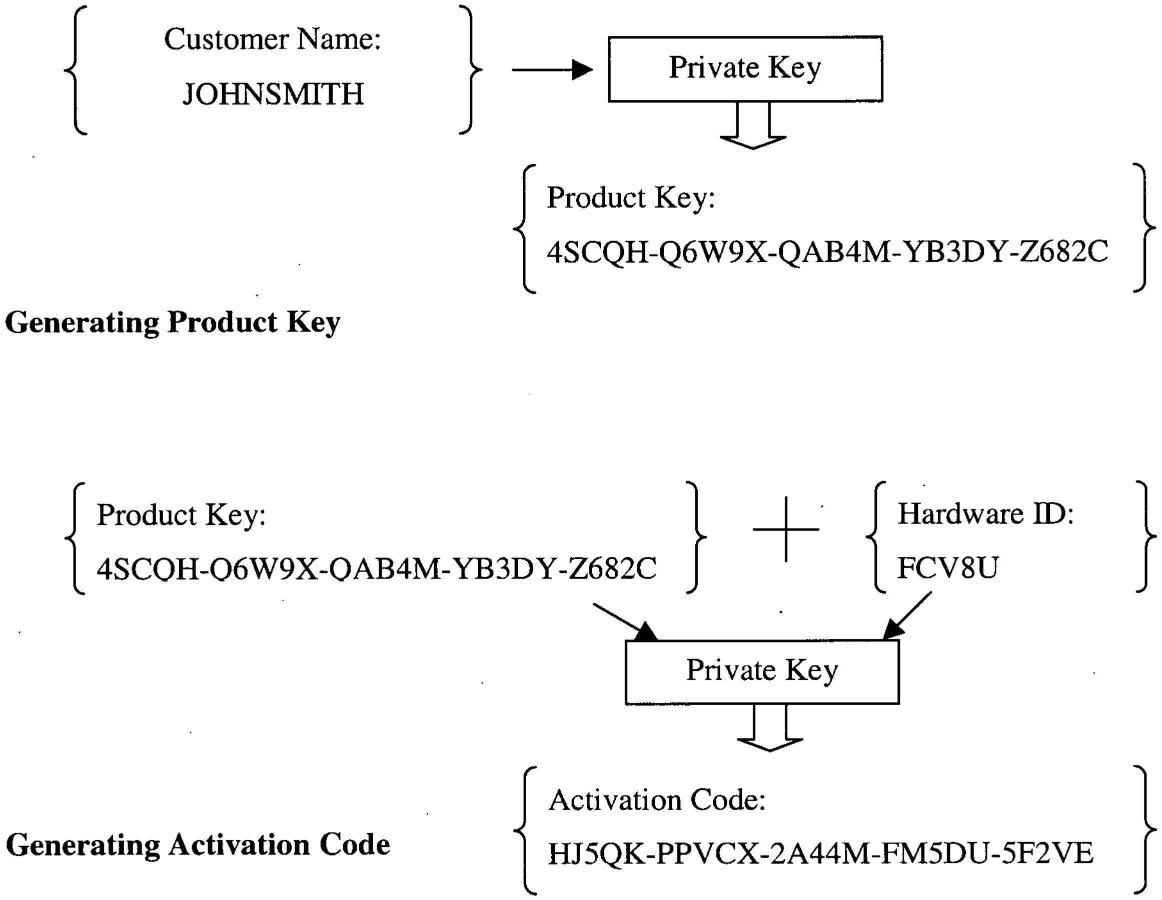
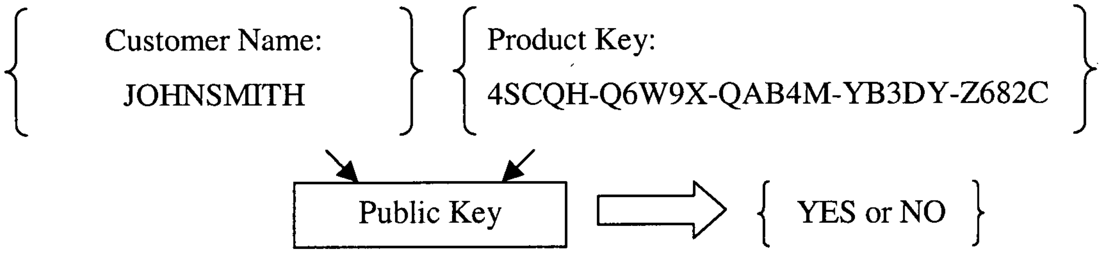
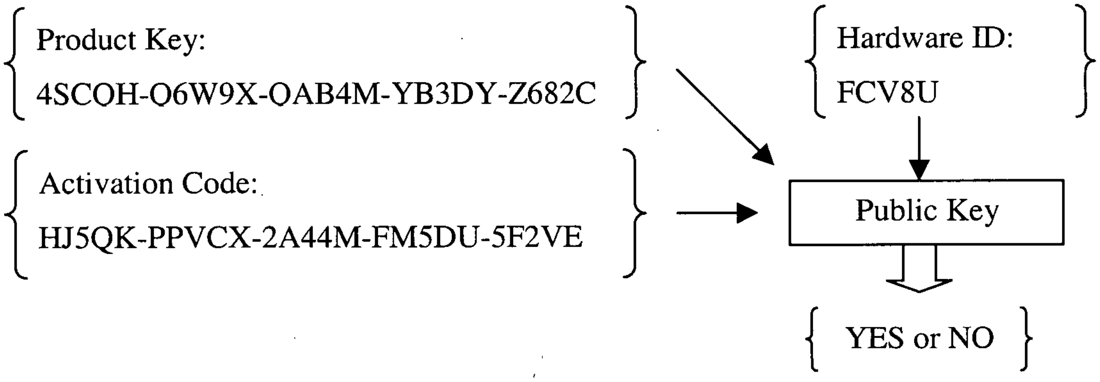


Figure 3 – Generation of Product Key and Activation Code

In order to verify a Product Key it is necessary to have the customer's name, its matching signature (the Product Key itself), and a *public key* that is stored in every copy of a product's executable. In order to verify an Activation Code, a Product Key, Hardware ID, and the same *public key* are necessary. An Activation Code will not be positively validated if a Hardware ID is not the same that was used to generate it, which means that either a product was transferred to another PC, or PC hardware was modified. In this case the product will require re-activation; our server will issue a limited number of Activation Codes per license, depending on a developer selected policy.



Validating Product Key



Validating Activation Code

Figure 4 – Verification of Product Keys and Activation Codes

3.1.3. Database

We made a decision to use Microsoft OS as an underlying platform for our system. Thus our natural choice of a database was Microsoft SQL Server 2000.

The first version of our system, developed in January – February 2002, did not include strong enough security measures; in particular, we allowed registered software developers to download their private keys; the idea was to allow developers to create licenses for their own software products locally on their own PCs. That turned to be a security flow that was once exploited. We made some other bad design decisions, such as using files to store private keys instead of encrypted memory streams. As a result, our system was hacked, and our registered developer's product key stolen by a malicious person. A key generator appeared for that product shortly.

The system used for this research was finished in April 2002. The bugs found in the first implementation were fixed, and the system design was changed to a multi-layer security model: we use registered COM objects as the only way to access a SQL database that physically resides on a server different from the one that has access to the Internet.

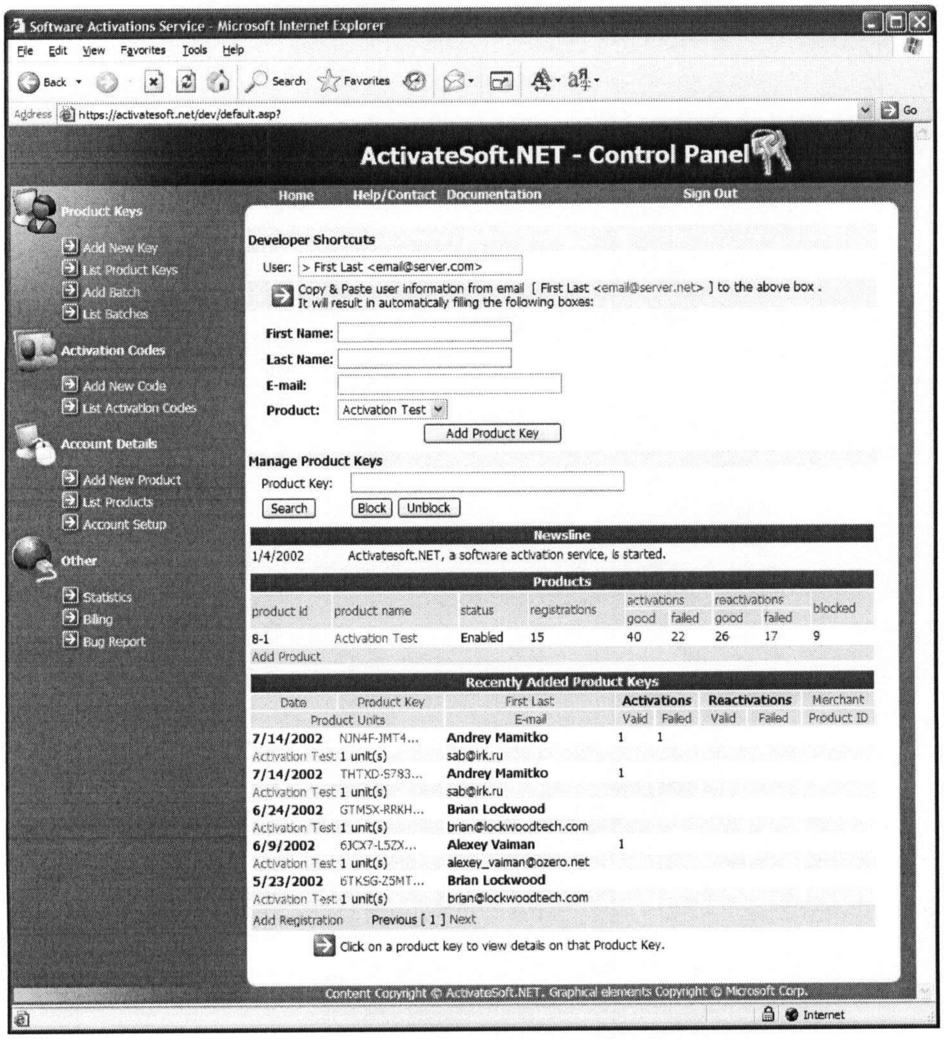
The database stores product keys, activation codes, and all developers' information; it contains encrypted private and public keys that are used for license generation; it also contains layers of code that enforce activation policies.

3.1.4. Online Interfaces and Developer's Control Panel

It was important to provide customers with a handy tool for activating their products. It was also considered a priority to have instant display and delivery of a Product Key at the time of ordering. Most other services provide a Product Key by e-mail; sometimes it takes a few days to get one after paying for a product. We added convenience by instantly displaying a Product Key on the Order page immediately after the customer placed an order and his credit card has been authorized. We can afford to do this because a key can be blocked online easily if the order turns to be fraudulent.

Our system uses SSL-encrypted connections with password authentication to produce Product Keys (passwords are unique for each entity authorized to issue Product Keys for a product, such as resellers and OEM distributors; in case one password is compromised, it will not affect other developers, products, or other issuing authorities).

Activation codes are delivered through a plain, non-encrypted HTTP connection that does not involve any authentication. This design decision was made because we do not consider any part of the information necessary for activation to be private or security sensitive. Only a Product Key along with Hardware ID is required to obtain an Activation Code; the Product Key can not be used to register software without customer's name in exact spelling. The customer's name or other personal information are never transmitted except at the time of ordering (order page is SSL protected).



https://activatesoft.net/dev/default.asp?

Developer Shortcuts

User: > First Last <email@server.com>

Copy & Paste user information from email [First Last <email@server.net>] to the above box . It will result in automatically filling the following boxes:

First Name:
Last Name:
E-mail:
Product: Activation Test

Manage Product Keys

Product Key:

Newsline

1/4/2002	Activatesoft.NET, a software activation service, is started.

Products

product id	product name	status	registrations	activations		reactivations		blocked
				good	failed	good	failed	
8-1	Activation Test	Enabled	15	40	22	26	17	9

Recently Added Product Keys

Date	Product Key	First Last	**Activations**		**Reactivations**		Merchant
Product Units		E-mail	Valid	Failed	Valid	Failed	Product ID
7/14/2002	NJN4F-JMT4...	**Andrey Mamitko**	1	1			
Activation Test 1 unit(s)		sab@irk.ru					
7/14/2002	THTXD-S783...	**Andrey Mamitko**	1				
Activation Test 1 unit(s)		sab@irk.ru					
6/24/2002	GTM5X-RRKH...	**Brian Lockwood**					
Activation Test 1 unit(s)		brian@lockwoodtech.com					
6/9/2002	6JCX7-L5ZX...	**Alexey Vaiman**	1				
Activation Test 1 unit(s)		alexey_vaiman@ozero.net					
5/23/2002	6TKSG-Z5MT...	**Brian Lockwood**					
Activation Test 1 unit(s)		brian@lockwoodtech.com					

Click on a product key to view details on that Product Key.

Content Copyright © ActivateSoft.NET. Graphical elements Copyright © Microsoft Corp.

Figure 5 – Developers Control Panel

The Developers Control Panel allows registered software developers to add and modify products, create Product Keys and Activation Codes, manage (block, unblock, change number of licenses and expiration dates, etc.) licenses, re-send Product Keys and Activation Codes to customers via e-mail, see statistics, and perform additional service functions.

3.1.5. C++ SDK

Modification of a product source code is required in order to enable the use of activations. In order to simplify and automate the process I have developed a SDK. The ActivateSoft.NET SDK is a set of C++ classes that should be added to a software project. The SDK allows recognizing our Product Keys and Activation Codes, activation over the Internet or without the Internet connection by entering an Activation Code manually.

The SDK contains verification module, our implementation of an open crypto algorithm described in section 3.1.2. It is necessary that a *public key* is built into a product; that public key must match a corresponding *private key* for that product. Private keys are stored on the ActivateSoft.NET server and used for Product Key and Activation Code generation.

The current version of the SDK only supports MS Visual C++. We are developing versions of the SDK for Borland Delphi and MS Visual Basic now. We also plan developing a 'wrapper' style SDK that will require no modifications to source code, and will protect any executable file in conjunction with a compatible third-party software protection tool such as ASProtect, *www.aspack.com*, or EXECryptor, *www.softcomplete.com*.

The SDK contains four layers of APIs. A developer can use any number of low levels of this API, and not use higher levels, if he prefers. Even the lowest layer contains all functions necessary to verify licenses and connect to the activation server. Higher levels add convenience, and include a default implementation of a GUI.

The first layer contains plain C functions that validate Product Keys and Activation Codes, load and store Product Keys and Activation Keys, store the date of product installation, and connect to the activation server. All information is stored in the Windows Registry under the

HKEY_CURRENT_USER key, which is the only key that is write-enabled for non-administrative users. If there are multiple users on the same PC, all of them will have to go through the activation process. The activation system will provide Activation Codes to all of them as these requests are coming from the same PC.

The product installation date is used to check whether a non-activated copy can be used the day it is started, or if it must be activated in order to continue functioning. This date is encrypted with a hardware-dependent password to prevent tampering.

The second and all subsequent layers are implemented as C++ classes. The second level API calls the previous layer functions. It contains simple C++ calls used to check if the product is registered and/or activated, and a function that connects to the activation server, transmits license information and performs activation.

The third layer contains graphical user interface functionality: an activation reminder dialog, a dialog for entering a Product Key, and an Activation Wizard (a set of dialogs guiding the user through the activation process). This layer uses the Microsoft Foundation Classes (MFC) library for GUI elements programming.

The fourth level has a single function `DoAllWork()` that handles everything from checking whether the product is activated or not to calling the Activation Wizard.

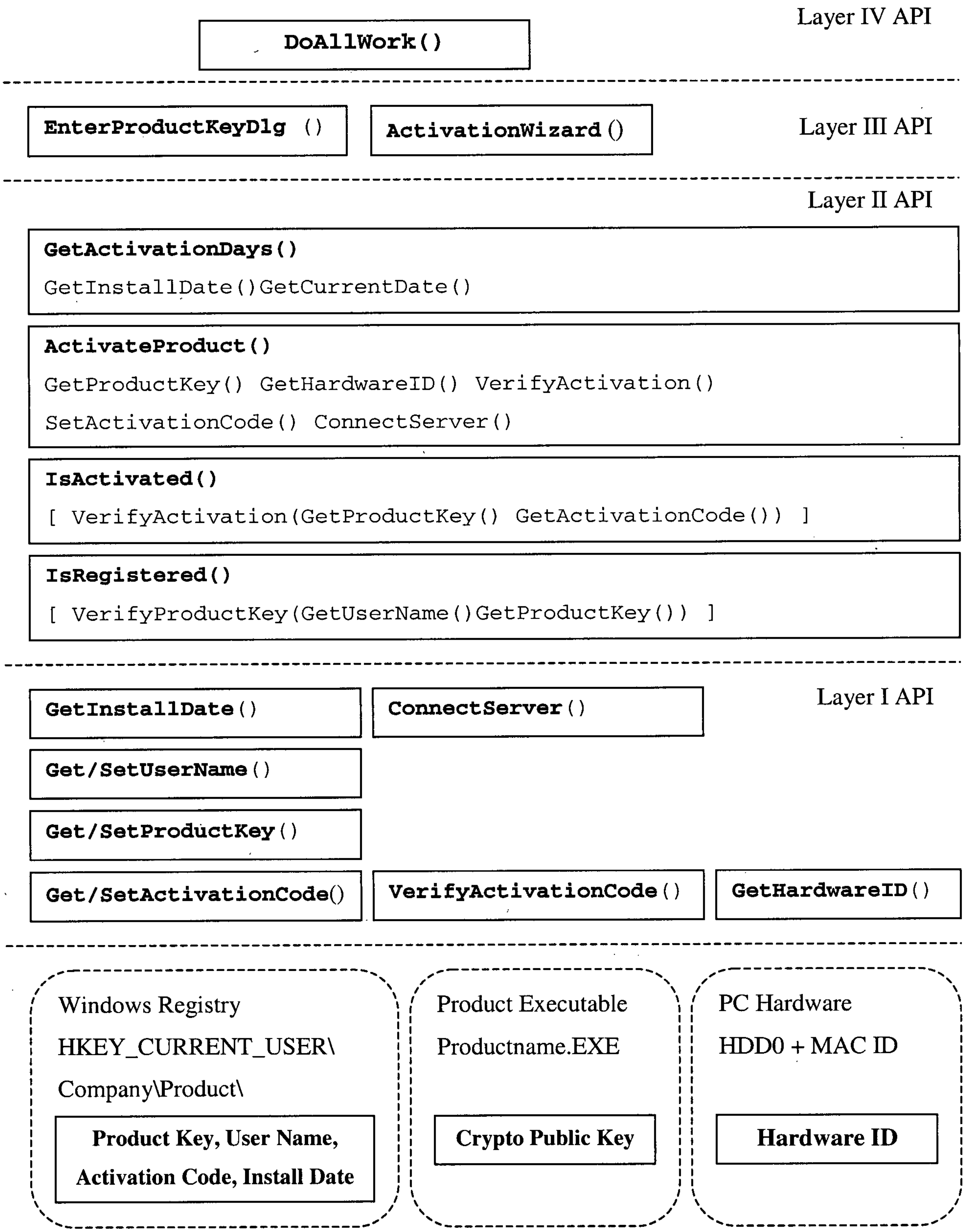


Figure 6 – Activation API

3.1.6. Developers of ActivateSoft.NET

The software activation system was developed by three parties:

- Pinrgam Marketing, a Vancouver based company;
- Indegro, an Irkutsk (Russia) based company;
- SoftComplete, a Ukrainian company.

Pingram Marketing (Oleg Afonin, Mikhail Dyachkov) provided the concept, funding, and requirements specifications.

Indegro (Anton Baranchuk, Mike Granin and Tim Loginov) developed an SQL database, ASP scripts and Web access to all the data, including Developer's Control Panel. All Web and server side programming is done by that company.

SoftComplete (Andrey Belokon) has developed a strong open-crypto algorithm for 25-characters secure Product Key and Activation Code generation, and implemented cryptography software for the system.

I created hi-level specifications, developed C++ SDK for developers, designed Developer's Control Panel that was then implemented in the system, participated in system deployment and testing, created usability specifications, and performed usability tests. It was my idea to create the service, ventured by Pingram Marketing.

3.2. Activation Policies

This section presents activation policies, pre-set types of system behaviour in relation to whether or not a user is allowed to activate. The section contains all technical details on how the system works during the experiment.

The system was designed to give all developers full control on activation policies. Initially we allowed each of the following selections for every software product supported by the activation system:

1. **Monitor only mode**. In this mode the system does not block any activation requests. It simply logs the requests to collect usage statistics.
2. **Massive fraud prevention only**. This mode only refuses to produce an activation code after a certain number of activation attempts are made from different computers. The product developer sets the number. A product key that was attempted to activate multiple times is marked as used illegitimately.
3. **License terms enforcement – fair use**. The system allows two activations per license purchased. Examples of 'fair use' are business and home PCs, a desktop and a laptop, and so on.
4. **Strict license terms enforcement**. This policy is targeted to expensive products. Number of permitted activations matches number of licenses; re-activations on the same PC are allowed.

If a user attempts to activate more licenses than were purchased, the system puts a temporary block on that license key. If a lot of activities happen such as multiple activation requests from different PCs, or overwhelming number of re-activation requests from the same PC, the system puts a "permanent block" on that license key.

The system controls activation policies of every product with a set of variables.

The "**Activation Period**" parameter controls the number of days after which the system clears temporary activation blocks from a license key, resets a list of PCs on which a license was activated, and resets all activation and re-activation counters. Permanently blocked licenses remain blocked.

The "**New Activations**" parameter controls the number of activations per license allowed during the activation period. If the system detects an attempt to activate the license, and the number of activations exceeds the number of "New Activations", the system will not produce an Activation Code.

The "**Re-Activations**" controls the number of activations allowed that originate from the same PC. It works similar to the "New Activations" parameter, except that it counts activation requests from

PCs that activated a license in the past. The only reason to use this parameter was to prevent server flooding by malicious users, thus avoiding extra CPU load and reducing number of writes to the database.

“**Fraudulent Use Threshold**” controls the number of failures such as failed activation requests, after which the system permanently blocks a license for fraud. This condition occurs if a user tries to re-activate the same product multiple times with a short delay (one day or less), or the system registers a large number of activation requests from different PCs.

The “Monitoring only” policy is implemented easily by setting the “activation period” to 0 days, in which case an activation counter will reset immediately after the activation. We implemented “Strict license term enforcement” policy by setting the “activation period” to 365 days and “new activations” to match the number of purchased licenses. Setting “activation period” to 30 and “new activations” to 2 per every purchased license conducts the “License term enforcement – fair use” policy.

The table demonstrates how different policies work.

	# Activations	# Reactivations	Period, days	# Threshold
Monitor only	unlimited	unlimited	0 days	unlimited
Massive Fraud Prevention	10	50	30 days	50
Fair Use	2	10	30 days	10
Evaluated Policy	3	10	30 days	10
Strict Enforcement	1	5	30 days	5

Table 1 : Activation Policies

I collected data for this research with the “Fair Use” policy, except that I allowed 3 activations per month instead of the default number of 2. I did it with research purposes in order to count the number of single-license users activating more than two licenses. “Fair use” was selected as a first estimation of the most convenient license enforcement policy based on a typical office software license that allows using the product on a home PC and at work, or on a desktop and a laptop.

4. Problems and Questions

This section describes problems and questions targeted in the thesis. The section lists assumptions made for conducting this research. It presents the main question of the research, and describes why this question is important to a group of software companies.

There are many different license control and license enforcement schemes. The one evaluated in this research is based on the online software activation technique. Our interest is whether this particular scheme gives any value to software development companies.

4.1. Key Question

The key question of this thesis is whether or not license enforcement with online software activation reduces software piracy by enforcing a number of PCs on which a software license can be installed.

I am going to try to shed light on this question by evaluating a license enforcement system and collecting data about how people use it.

The question is important because there are many small software companies that will benefit from this research, which allows them to make better decisions selecting the software activation parameters for their products, thus, on one hand, maximizing the degree of software license protection, and, on the other hand, reducing the number of complaints and technical support inquires from their customers, and increasing customer satisfaction.

4.2. Intermediate Questions

In order to answer the main question and make a final conclusion on whether or not the system can be profitable to small software developers, a better understanding of user behaviour must be achieved. To better understand the problem, I set three intermediate goals:

1. Determine how many different machines on which users attempt to activate the software. It is important to figure out whether a common practice is installing a single license on a single machine, or it is a common practice to regularly install on more machines than the number of license installations.

   Question #1 gives better understanding of the habits of legitimate customers who can in theory purchase additional licenses when they are denied activation.

2. Determine how the system affects legitimate users. It is important to figure out if the customers trying to activate more licenses than they have purchased will purchase additional licenses or not, and if a significant number of customers will be annoyed by the fact they have to activate to a degree that they refuse using the product in favour of a similar product that lacks activation.

   Question #2 provides better understanding of possible disadvantages of activation license enforcement, measured as the number of annoyed customers who will not use the product or purchase additional licenses solely because of the license enforcement.

3. Determine a number of users illegally passing their product keys to other users. This number includes pirates who publish license codes on the Internet on a variety of crackers' Web sites. By comparing information collected answering questions #1 and #3 I will try to estimate the ratio of *legitimate* and *illegitimate* installations of a software product.

   Question #3 allows to see those users who are not legitimate customers; I assume this category of users not only will never pay any money for a license (their purchases often turn into charge backs), but also distribute their licenses over the Internet, making a software development company lose potential customers and increasing expenses on Web site traffic.

4.3. Assumptions

To answer the questions I make several assumptions.

First, I assume that new and upgrade users are discriminated (we sent them license for a product with a different name, such us "Product A Upgrade"). "Upgrade users" are those who purchased and used one of the previous versions of a product, with no software activation, and upgraded or did not upgrade to the new version with software activation. In order to have bigger sample we offered them a free upgrade by emailing messages containing their new Product Key, and instructions on how to upgrade.

My second assumption is that neither upgrade nor new users know a priori about license enforcement in the new version of the product they are offered to upgrade to. As our test company is small, "word of mouth" is negligible in this case. This may be important not to affect their decision to upgrade to a new version. If they knew about the activation before upgrading, it might've affected their decision on whether they want to upgrade because they do not like products tying itself to hardware, no matter how simple the process may be. This way I tried to reduce at least one factor preventing existing customers from upgrading, and increase sample size.

4.4. Data Collected by the Activation System

This section explains what data our system has access to.

ActivateSoft.NET monitors and stores all user activity. This data can be processed later. We were collecting the following data from April 15, 2002 to September 25, 2002.

1. Whether or not this is a new purchase or an upgrade
2. All registration and activation details are collected
    a. Date of transaction
    b. User's name and email
    c. IP addresses

   d. Originated from Web or application
   e. User's Hardware ID
   f. Number of new activations and reactivations

3. Statistics:
   a. Number of successful new activations.
   b. Number of successful re-activations.
   c. Number of failed new activations.
   d. Number of failed re-activations.
   e. List of successfully activated Installation IDs
   f. List of failed new activations (Installation IDs)
   g. List of failed re-activations (Installation IDs)
4. Support enquiries regarding the activation (e-mail messages)

User's name, email, date and time of purchase, and the Product Key are collected at the time of ordering. The rest of the information is collected during the activation.

This data is necessary in order to find an answer to the key question.

4.5. Limitations of Study

There are several things that must be taken into consideration while performing this research.

First, our sample is fairly small. Due to the time constraints and the fact that the system had to be developed prior to collecting the data, we were able to collect data on software activations 15 April to 25 September, with two products on sale. During the experiment a total of 997 licenses were issued, including 359 newly purchased ones and 638 distributed for free as an upgrade to existing users. Out of those, 544 licenses were activated; the reasons why other users preferred not to activate are beyond our knowledge.

I collected customer's feedback from April 15, 2002 to September 24, 2002, and received 112 messages with different comments on the activation process, out of which 23 were complaints.

The main limitation is that with the data we are able to collect we do not know the reasons of several customer behaviour patterns. We are only able to collect a limited set of data due to commercial issues, design faults and technical limitations. For example, we could technically ask customers attempting to activate a product on an additional PC for the reason they do that by providing a multiple choice questionnaire or free text input field, and make this a requirement in order to get an additional activation code.

Our data does not provide answers to behavioural questions; neither has it provided information on the reasons why they do one thing or another. The data, however, contains information on the numbers: how many users, how many PCs, how many activations, and so on.

When customers get an upgrade license (upgrade licenses were distributed for free) but never activate it, the customers may not actually use the product, or they will activate only when they are forced to by the activation timer (in our implementation there is a 14-day allowance), or they will revert to a previous version of the product (which is also possible and happens regardless of whether or not an activation scheme is used).

When a customer purchases one license and tries to activate on multiple machines, it may be either illegitimate behaviour (trying to install on a number of machines greater than allowed by the license agreement), or legitimate behaviour (upgrading a machine fairly often). Chances of the second case are low because our system uses a Hardware ID that is least likely to change during the upgrade.

Most of the questions could be answered by conducting a targeted survey among the registered users. However, I do not have skills and resources for successfully conducting such survey.

5. Evaluation

This chapter explains how the experiment was run, describes two software products on which activation performance was evaluated.

5.1. Experiment Setup

The experiment was run from 15 April to 25 September 2002. I decided to evaluate a policy that is as close as possible to Microsoft policy on their Windows XP activation (even if it is not optimal, it's good to begin with). The default activation policy is "Fair Use", activations allowed on 2 different PCs per month. I modified the default "Fair Use" policy to allow 3 unique activations per month instead of 2; I did that to learn more about the number of unique PCs users install their licenses on.

The experiment will answer the first question of this research, about the number of copies of the program a legitimate user, who purchased one license, installs on different machines. I propose evaluation of the "fair use" licensing policy to shed light to the third question of this research, the number of users who illegally pass their product keys to other users. This policy assumes two computers to be a 'fair use' (such as home and work computers, or a workstation and a laptop). The second question, about how the system affects our legal users, is to be answered by analyzing customers' feedback, particularly problem reports.

In order to collect and analyze this data we have all raw data appropriately analyzed. At the end of our test period all necessary data was collected. After that I analyze the data in order to figure out answers to the questions asked earlier in this research.

Using this "fair use" policy and data collected I also hope to determine what *fair use in case of shareware is* in terms of number of licenses a majority of legitimate users need when purchasing a single license. I run the experiment under assumption that a 'fair' user needs two licenses per activation period (i.e. home and work PCs, or desktop and laptop computers). I also suggest that it is fair enough to reset activation data every 30 days as the user may change or upgrade PCs.

5.1.1. User's Technical Support Inquiries

We received 112 messages from the users of our programs who encountered problems with or had comments about the activation system. 23 of these messages were complaints. For each of those users we have complete information about when they've purchased the product, whether they are upgrade or new customers, and so on.

5.2. Products Used for Evaluation

This section presents details about the experiment: participating company and two of its software products as it was prior to using the activation system and after the company started using it.

5.2.1. Products

During the experiment I analyzed statistical data of the evaluated company (which name is not disclosed), a software development company that has two products: *Product A* and *Product C.*

Product A ($49.95 US new license, $14.95 additional license, free upgrades from previous version) is fairly old and popular. It was originally targeted to home users only, yet later some functions for home offices and business use were added. The majority of its purchasers are private users and home offices (about 85% according to our data), and only about 15% are companies. Unfortunately, we are unable to distinguish between the home offices and private users at the moment.

Product C ($29.95 US new license, $9.95 additional license, free upgrades from previous version) was designed later and was targeted mainly to small businesses and home offices. About 30% of those who purchased Product C are businesses; another 70% are home offices and private individuals.

5.2.2. Upgrade Users

	Licenses Issued	Licenses Blocked	Licenses suspended	Activations	Reactivations	Attempts to (re)activate blocked licenses
ProductA	461	5	0	296	67	45
ProductC	177	0	0	93	161 *	0

Table 2 – Upgrade users' activation figures

Table 2 contains numbers of licenses issued to upgrade users.

5.2.3. Newly Purchased Users

	Licenses Issued	Licenses Blocked	Licenses Suspended	Activations	Reactivations	Attempts (re)activate blocked licenses
ProductA	274	11	1	341	106	79
ProductC	85	4	0	101	78 *	15

Table 3 - New users' activation figures

Table 3 contains number of new customers who purchased the products, and number of times the licenses were activated.

Note that Product C has an abnormally high number of reactivations. It happened because initially the product was released with an older version of the Activation SDK that contained a bug preventing it from saving an activation code on the local machine, requiring re-activating every time the product was started. The bug was fixed in about a week. The bug only affected the number of reactivations, and did not affect the number of activations and blocked keys; I considered that it is still possible to use this data to answer the questions of this research.

5.3. Data Collected During the Experiment

This section presents data in the form of tables that was collected by the activation server during the experiment. This data is used to answer our intermediary questions.

5.3.1. Number of activations per license

This chapter contains multiple tables with data on the use of licenses. The left column of each table contains the number of activations per license key. The right column contains the number of licenses that were activated the number of times shown in the left column.

All products are licensed with a modified ‘fair use’ enforcement policy that allows for 3 activations during the 30-day period. Once every 30 days they can activate a product on 3 more PCs. I issued additional activation codes if a user asked for it in order to figure out the best ‘fair use’ or ‘common use’ practices.

5.3.1.1. SUCCESSFUL NEW ACTIVATIONS

This section provides information that leads us to the answer to the first and second intermediate questions of this research, as well as provides a hint to answering the key question.

Table 4 shows the number of licenses that were successfully activated on multiple machines. The number of activations on different PCs is represented by the value of the first column in the table (N). The second column contains the number of licenses that were activated N times. This table gives a good hint on how many PCs an average user installs a product.

Product A – New Users	
# activations (N)	**# licenses activated N times**
0	41
1	164
2	46
3	12
4	7
5	3
6	1
Total Licenses	274
Activated Licenses	233
Product A – Upgrade Users	
0	262
1	130
2	51
3	9
4	7
5	1
6	1
7	
Total Licenses	461
Activated Licenses	199

Product C – New Users	
# activations (N)	**# licenses activated N times**
0	18
1	49
2	9
3	6
4	1
5	1
6	
7	1
Total Licenses	85
Activated Licenses	67
Product C – Upgrade Users	
0	132
1	28
2	3
3	5
4	6
5	1
6	1
9	1
Total Licenses	177
Activated Licenses	45

Table 4 - Number of activations per license: Successful New Activations

The licenses that were never activated are represented with '0' in the left column. This number is larger for upgrade users, which means that either they abandoned the product or that they use an older version and refuse to switch to the newer one. Besides the licenses that were never activated, the total number of licenses activated at least once is 100%; then we'll get the following normalized tables.

Product A	New Users (233 activated licenses)	Upgrade Users (199 activated licenses)
Activated once	70.39%	65.32%
Activated 2 times	19.74%	25.62%
Activated 3 times	5.15%	4.52%
Activated 4 or more times	4.72%	4.54%

Table 5 – Product A: activations per license, percentage

Product C	New Users (65 activated licenses)	Upgrade Users (45 activated licenses)
Activated once	73.13%	62.22%
Activated 2 times	13.43%	6.66%
Activated 3 times	8.96%	11.11%
Activated 4 or more times	4.48%	20%

Table 6 – Product C: activations per license, percentage

21.18% of all customers have never activated Product C after purchasing. This again might be explained by suggesting that the customers do not run this product every day.

Out of those who have activated their licenses, a majority have only done it once. Some number of users activated 2 times, and a few did it 3 times or more on different machines. A "fair use" policy that allows 2 activations a month would satisfy 68.88% of Product C users and 86.56% of Product A users. The test policy that I used for this evaluation, which allows 3 activations per month, would satisfy 95.5% of all customers.

The picture changes, however, for a one-month case. The following two tables compare May, June, July, August and September (numbers represent activations combined for new users of Product A and Product C) as the only months we have complete data for. April is excluded because I only have data for 2 last weeks of the month.

# Activations (N)	**licenses activated N times**	**licenses activated N times**	**licenses activated N times**	**licenses activated N times**	**licenses activated N times**
0	10	7	10	13	14
1	44	30	43	46	43
2	10	8	11	10	5
3	3	3	4	4	2
4	2		2		
5	1	1			
> 5			1		
# Product Keys	70	49	71	73	64
# Keys Activated	60	42	61	60	50
	MAY	**JUNE**	**JULY**	**AUGUST**	**SEPT**

Table 7 – Monthly Successful Activations

This data demonstrates the number of different computers on which a single license was activated during a 30 day period.

	May (60 activated licenses)	**June (42 activated licenses)**	**July (61 activated licenses)**	**Aug (60 activated licenses)**	**Sep (50 activated licenses)**
Activated once	73.33%	71.43%	70.49%	76.66%	86%
Activated 2 times	16.66%	19.05%	18.03%	16.66%	10%
Activated 3 times	5%	7.14%	6.56%	6.66%	4%
Activated 4 or more times	5%	2.38%	4.92%	0%	0%

Table 8 – Monthly Product Activations, Percentage

Data in this table demonstrates that users tend to use their software licenses on more machines over time (September data shows that). Table 8 shows that there is a significant number of customers who install software licenses on multiple PCs.

Data from this table leads us to the answer to the first question of our research: determining number of PCs a typical customer may need to use a single license on. The table allows me to conclude that, while many the users are only activating it on a single PC, some significant number (13% to 17%) of customers use it on 2 PCs, and around 5% - 10% uses it on 3 or more PCs. From this data I also suggest that the numbers of customers who use our test software on more than 3 PCs are negligible.

5.3.1.2. FAILED: ACTIVATION LIMIT EXCEEDED (TEMPORARY BLOCK)

This subsection gives a lead to answering the second question on system's influence to the customers.

"Temporary Block" condition is set when a user attempts to activate the license more times than he or she is allowed to do according to the terms of agreement. This condition is very different from the "Permanent Block"; temporary blocking will be removed at the end of the accounting period (in our test case this period is 30 days), while the permanent block won't be removed.

A temporary block is put on a license when a customer attempts to activate it on one more PC than he is allowed to. Temporary blocks are removed once every 30 days; however, if the number of activation attempts exceeds the value of "Fraudulent Threshold" parameter, the system permanently blocks this license.

Our system only provides information on licenses that were temporarily blocked in the current time period.

7 users of Product A were denied activation after exceeding their activation limit in the period of 25 Aug – 25 Sep. 3 Product C users were denied activation and put on temporary block.

I noticed a frequent pattern when users try activating a license key on an additional PC, and the activation system fails to produce an activation code with a "Number of allowed activations exceeded" message and its comprehensive explanation. The users still try to activate the license some more times in hope that there was a technical problem, but the system still does not return an activation code. They usually try 2-7 times before giving up. After that they have several choices:

contact us to ask for an additional activation or report a ‘technical problem’ with the activation system; purchase an additional license paying the low upgrade fee; or just waiting 30 days and attempting activation again.

During the study we received 7 requests to issue additional activation codes for Product A and Product C. The customers argued: “I have a third (fourth, etc.) PC at home and need your product to run on it”. In all cases we preferred to issue an activation code without arguing, although we didn’t have to according to the terms of our end-user license agreement.

3 times the users who were denied activation purchased upgrade licenses immediately, one for Product A and one for Product C.

5.3.1.3. FAILED: LICENSE IS BLOCKED (PERMANENT BLOCK)

This section contains data that is necessary to answer the third question of this research on the number of customers who illegitimately pass their licenses to other users.

Blocked licenses represent attempts of massive software theft. If a license was blocked by our system, then there were multiple attempts of unsuccessful activations from *different* machines. This may mean that a license was purchased with a stolen credit card and then uploaded to a ‘warez’ Web site; or that a license was purchased with a legitimate credit card, and then shared with several other users or installed on multiple machines in violation of the terms of license agreement; or some other violation of license terms.

- Product A, New Users
    - 1 license blocked, 25 activation attempts from unique PCs
    - 1 license blocked, 11 activation attempts
- Product A, Upgrade Users
    - 1 license blocked, 44 activation attempts from unique PCs
    - 1 license blocked, 35 activation attempts
- Product C, New Users
    - 1 license blocked, 4 activation attempts from unique PCs
    - 1 license blocked, 11 activation attempts

Table 9 : Number of activations per license: Failed – License Is Blocked Permanently

There are 4 Product A licenses and 2 Product C licenses that were blocked, which corresponds to the products nature: Product A is marketed primarily to home users, while Product C is designed for small business and home office use.

Each license was placed on a temporary block first (after 3 unsuccessful activation attempts), and then put to a permanent block (after another 10 unsuccessful activation attempts, according to our policy).

One license was blocked with a total of 25 activation attempts made from 25 different computers; another one received 44 activation attempts from different machines.

I assume that all licenses that were blocked were used illegitimately. We have never received complaints from those who purchased these licenses. In any case, this data leads us to the answer to the <u>third question</u> of this research, about how many customers are passing their licenses illegitimately to other users.

As can be seen from the data shown in Table 2 and Table 3, there are more blocked licenses than listed in Table 9. An explanation is simple: some licenses were blocked manually by the developers of Product A and Product C for various reasons, such as: a license was issued but a customer never paid; an order was turned fraudulent, and a charge back was issued; or an order was refunded.

5.3.2. Users' Technical Support Inquiries

This subsection presents information necessary to answer the second question about the system's influence to users who we consider to be legitimate.

I collected customer's feedback from April 15, 2002 to September 24, 2002, and received 112 messages with different comments on the activation process, out of which 23 were complaints.

All complaints report different problems activating the software. The company received 23 complaints from the customers. Most of the messages were related to products developed with earlier versions of Activation SDK that contained a bug preventing activations in certain conditions. Once the SDK was fixed, we stopped getting this kind of complaints from the users.

A very common reason for activation failure is a presence of some local proxy server or firewall. In that case we advise the users to use a Web browser in order to activate a product. Some minor problems (mostly connected with initially poor GUI and lack of documentation) were expected and happened (there were people who tried to activate their products 15 times before contacting support). This kind of problems had not occurred since last month, when some GUI changes were made in the system, and better help was added to the Activation Wizard.

Another reason for a failed activation is exceeding the maximum number of allowed installations. Many users use software they buy on more than one machine. The company tolerates one or two additional installations with the "Fair Use" activation policy; however, some customers require more than that, and usually are denied activation. There were 4 support requests of this kind.

During the test period, the company provided 7 users with additional licenses for free. 3 customers paid for additional Product C licenses, and one additional upgrade license was sold for Product A.

We did not receive any 'privacy concern' feedback, despite the fact we expected it. Those concerned about their privacy on the Internet do have some kind of firewall blocking Internet access to applications like Product A. They usually open a Web browser manually, and enter all required information. As we do not ask for any personal information in order to activate a product, they are usually satisfied with our privacy commitment.

5.4. Summary of Evaluation

This subsection summarizes results of the experiment.

A total of 997 licenses were issued during the experiment, including 359 newly purchased ones and 638 distributed for free as an upgrade to existing users. 544 of these licenses were activated; the other users preferred not to activate for an unknown reason. Most were distributed for free as an upgrade to existing customers; and thus the most likely explanation is that they never wanted to update. 59 licenses were never activated by 359 new customers who paid for a license, while 394 never-activated licenses were abandoned by 638 upgrade users who received it for free.

Averaging data collected during May, June, July, August and September 2002 (a total of 327 licenses, 273 of which were activated at least once), ~75.45% of those who activated a product did it on a single PC during one month since their first activation; few users (~16.12%) installed it on two machines during the same month; a number of customers (~8.43%) installed it on three or more PCs during the same month.

Analyzing a larger sample (period of April 15th to September 25th, 2002, 359 total new licenses and 300 activated licenses), 71% of customers activated a product on 1 PC, 18.3% - on 2 PCs, and 10.66% activated on 3 or more PCs (data for new customers buying their first license). Similar numbers for upgrade users who got their licenses as a free upgrade: 64.75% of customers activated a product on 1 PC, 22.13% - on 2 PCs, and 13.12% activated on 3 or more PCs.

2 licenses were blocked and believed to be tampered with / illegally distributed (which counts to 0.45% of pirated licenses) to the day of August 25th, 2002, and 4 blocked tampered licenses (0.74%) to September 25th, 2002.

1 license was blocked with 25 unsuccessful activation attempts from different PCs; another one received 44 activation attempts from different PCs. 2 more licenses received 35 and 11 unsuccessful activation attempts, making a total of 125 possible illegitimate users trying to violate the terms of a license agreement. The number looks significant in relation to the total of 544 active users.

We received 23 complaints about issues, all of which positively resolved during the period of 15 April – 25 September 2002.

Referring to our key question about the effectiveness of software activations to reduce software piracy, about 10% to 13% of all customers use more licenses than they are allowed to by ‘Fair Use’ software license (‘Fair Use’ policy allows 2 installations on different PCs), and about 4.5% of customers would violate the policy that was used for the research that allows 3 installations of a license. Approximately 30% of Product A and 37% of Product C customers would violate conditions of the ‘Strict’ enforcement policy (allowing a customer to install a product on a single PC).

It is difficult to estimate an exact value of the fact that the four licenses were blocked. Those users who were denied activations are unlikely to purchase a legitimate license. Those license codes, however, might have been widely distributed on some cracker’s site, and, based on our previous experience, result in thousands of people downloading that codes and using them illegitimately with the products.

The company had experienced exactly same situations several times. First time it happened the sales were down significantly (about 30% down during the month when a stolen license was published) because the company did not know how to react properly. Later on they managed to block licenses in the program itself (uploading a modified version to the Web site), thus minimizing the effect of those stolen keys.

6. Related Work

This chapter describes references to related work, and compares them to this research.

Microsoft Corp., as a developer of Windows Product Activation, is naturally the primary source of information about the activation.

Microsoft Piracy Basics [8] contains general information on what software activation is, and explanation of how it works, including the technical details, frequently asked questions about activation, their Privacy Policy on activation, and more. Their Frequently Asked Questions [12] was so useful that I included some quotations from there to my thesis in order to explain the concept of activation.

Microsoft Technical Bulletin on Activation [9] was the source of information for designing a system for this evaluation. The article provides detailed description of Microsoft's product key, Installation ID, and activation codes format, with an overview of corresponding digital certificate and cryptography technologies. *"Microsoft believes that product activation will be successful at deterring the casual copier, thereby reducing the piracy of Windows XP. Product activation achieves this goal by implementing a technology solution that deters the casual copier [...]"* [9], they conclude.

Von Mike Hartmann in [5] reverse engineered Microsoft's WPA implementation, described the technical details he found out, and published C source code as a sample of technologies used for the WPA implementation. *"The Windows Product Activation (WPA) that is implemented in the current RC1 of Windows XP shows some serious bugs which will open the way for hackers to avoid the whole system"* [5], he claims. The paper concentrates on low-level analysis of the system code.

Joe Wilcox, a CNet editor, discusses in [23] issues of privacy and fair use connected to this new technology. *"The company's new product-activation technology, which locks Office XP or Windows XP to a particular PC hardware configuration, can deactivate unexpectedly, rendering the software useless until a code number is obtained from Microsoft. The feature could present the* *biggest headache to people that frequently upgrade or change components on their PCs"* [23], he argues. He collected and published negative feedback of some users who were turned down by Windows Product Activation and decided to switch to an alternative platform because of that. Many similar articles are published regularly on the Web.

John C. Dvorak discusses in PC Magazine article [16] copy protection issues, and criticizes the Microsoft approach to copy protection via software activation. The author points out that it is unknown how many times one can install Windows XP, or whether it can be remotely disabled: *"How many times can you actually activate XP? The number three comes up a lot when I look for answers on this topic, although there is no definitive answer on Microsoft's Web site."* [16]

Issues he discusses in the article were the reason why we disclose and precisely describe our activation policies to all users of our activation system. *"XP is a marvellous OS—a workhorse that everyone should use. But it hasn't set the world on fire and the muddy and vague information about how activation works and its limitations are why the sales have not been stellar."* [16]

7. Conclusion

I designed ActivateSoft.NET, an online software license enforcement system that enforces terms of a product's end-user license agreement, which was implemented by developers from several companies [section 3.1.6]. I developed "fair use" policy for that activation system, a policy that allows customers to install a product on 2 different machines during a 30-day period, and performed a study on system behaviour using real world software products for 6 months (15 April to 25 September 2002).

The activation system made it possible to monitor and control the way the customers use their software licenses. I determined a 'fair use' policy that allows installations on 3 different PCs per month would fit the products used for the research better than other policies because it satisfies approximately 95.5% of all customers.

The research answered the <u>first question</u> about a number of copies of a product a legitimate user with a single license installs on different machines. Averaging data collected during May, June, July, August and September 2002 (a total of 327 licenses, 273 of which were activated at least once), ~75.45% of those who activated a product did it on a single PC during <u>one month</u> since their first activation; few users (~16.12%) installed it on two machines during the same month; a number of customers (~8.43%) installed it on three or more PCs during the same month.

Analyzing a larger sample (period of April 15th to September 25th, 2002, 359 total and 300 activated licenses), 71% of customers activated a product on 1 PC, 18.3% - on 2 PCs, and 10.66% activated on 3 or more PCs (data for new customers buying their first license). Similar numbers for upgrade users who got their licenses as a free upgrade: 64.75% of customers activated a product on 1 PC, 22.13 % - on 2 PCs, and 13.12% activated on 3 or more PCs.

This research provided the answer to the <u>second question</u> regarding the effect of the system on average users. We received 23 complaints about the activation; the low number of complaints allows me to <u>conclude that the system does not negatively affect legitimate customers</u>.

A number of users who illegitimately shared their licenses and made their product keys accessible to other users are 6. I do not consider this to be a big enough sample, thus can not give a definitive answer to the third question. The number of 6 tampered licenses can only give a hint to answering the question.

Direct effect on piracy is measurable and significant. 6 licenses were stolen; 139 attempts were made to activate them on unique machines by non-authorized users. Indirect effect on piracy is significant: based on our previous experience, uncontrolled distribution of illegally obtained licenses hurts sales. This impact is more significant for home-user oriented products, and less significant for business-user oriented products. I believe that the use of the activation system alone has prevented those stolen licenses from being published on crackers' Web sites.

An important hint to answering the key question of this research referring to the effectiveness of software activations in reducing software piracy, that about 10.66% of customers are using more licenses than they are allowed by the 'Fair Use' policy that allows installing a product on 2 different PCs. It is also significant that approximately 29% of customers would violate conditions of the 'Strict' enforcement policy allowing them to install a product on a single PC only, which creates potential of making additional sales to these customers. Due to the small sample size the numbers lack precision.

I provided a good hint answering the key question of this research. The research allows each software developer to decide whether or not using online activation technique will be cost effective and commercially profitable to their particular products, and provides information on the effect that such a system has on the customers.

Bibliography

[1] Digital-Ticket-Controlled Digital Ticket Circulation

Ko Fujimura, Hiroshi Kuno, Masayuki Terada, Kazuo Matsuyama, Yasunao Mizuno, and Jun Sekine,NTT Information Sharing Platform Laboratories

http://www.usenix.org/publications/library/proceedings/sec99/fujimura.html

USENIX Technical Program - Abstract - Security Symposium 1999

[2] Detecting and Countering System Intrusions Using Software Wrappers

Calvin Ko, Timothy Fraser, Lee Badger, and Douglas Kilpatrick, NAI Labs, Network Associates, Inc.

Abstract - Security Symposium – 2000

http://www.usenix.org/publications/library/proceedings/sec2000/ko.html

[3] Bitwise Operator: The Plain Truth About Piracy. By fprefect

http://www.ambrosiasw.com/cgi-bin/ubb/newsdisplay.cgi?action=topics&number=14&article=000052

[4] Why Do People Register, Does Crippling Work, Does Anybody Really Know? By Colin Messitt

http://hackvan.com/pub/stig/articles/why-do-people-register-shareware.html

[5] Windows Product Activation compromised (Inside the WPA)

VON MIKE HARTMANN

http://www.tecchannel.de/betriebssysteme/746/index.html

[6] Microsoft Product Activation Overview

May 30, 2001

http://www.microsoft.com/office/evaluation/indepth/activation.asp

[7] Technical Details on Microsoft Product Activation for Windows XP

http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/xpactiv.asp

[8] Piracy Basics: Microsoft Product Activation Overview

http://www.microsoft.com/piracy/basics/activation/

[9] Technical Market Bulletin on Product Activation in Windows XP

http://www.microsoft.com/piracy/basics/activation/windowsproductactivationtechnicalmarketbulletin.doc

August, 2001

[10] Deloitte & Touche product activation audit report

http://www.microsoft.com/piracy/basics/activation/microsoft_opinion_report_english.pdf

October 3, 2001

[11] Windows XP Product Activation

http://www.microsoft.com/windowsxp/pro/evaluation/overviews/activation.asp

June 29, 2001

[12] Microsoft Product Activation: Frequently Asked Questions

http://www.microsoft.com/piracy/basics/activation/mpafaq.asp

[13] How XP WPA will squeeze more money out of businesses

http://www.theregister.co.uk/content/4/20912.html

09/08/2001

[14] Software protection for shareware developers

Steve Lower - Chem1Ware Ltd. - steve@chem1.com, Educational Software Cooperative Newsletter, (Volume 9 No.7), December 2000.

http://www.sfu.ca/person/lower/MISC/softprot.html

[15] Piracy Basics - Microsoft Product Activation: Windows XP Service Pack 1 Changes to Product Activation

http://www.microsoft.com/piracy/basics/activation/windowsxpsp1.asp

[16] Straight Talk on XP Activation

By John C. Dvorak

PC Magazine, August 12, 2002

http://www.pcmag.com/article2/0,4149,462414,00.asp

[17] The Wrong Approach to Copy Protection

By Michael J. Miller

PC Magazine, April 2, 2002

http://www.pcmag.com/article2/0,4149,6312,00.asp

[18] Discrete logarithms in finite fields and their cryptographic significance.

A. M. Odlyzko

[19] Discrete logarithms: The past and the future.

Andrew Odlyzko

[20] A Study on the Proposal Korean Digital Signature Algorithm.

Chae Hoon Lim, Pil Joong Lee

[21] Lattice Reduction: a Toolbox for the Cryptanalyst.

Antonie Joux, Jacques Stern

[22] A Key recovery Attack on Discrete Log-based Schemes using a Prime Order Subgroup.

Lim & Lee, Crypto '97

[23] "Microsoft's XP: Hardware changes a turnoff" by Joe Wilcox published on CNet News

(*http://news.com.com/2100-1001-269085.html?legacy=cnet*)

Appendix 1. Microsoft Windows Product Activation FAQ

What is Product Activation?

Microsoft Product Activation is an anti-piracy technology designed to verify that the product has been legitimately licensed.

How does Microsoft Product Activation work?

Product Activation works by validating that the software's product key, required as part of product installation, has not been used on more PCs than is allowed by the software's license. Product key information, in the form of the product ID, is sent along with a "hardware hash" (a non-unique number generated from the PC's hardware configuration) to Microsoft's activation system during activation. Activation is completed either directly via the Internet or by a telephone call to a customer service representative. Activations on the same PC using the same product key are unlimited. Product Activation discourages piracy by limiting the number of times a product key can be activated on different PCs.

Will Product Activation make it more difficult for customers to install and use the software?

Product Activation is designed to be simple and unobtrusive for customers who legitimately acquire the software license. Customers will have the choice of activating via the Internet or by telephone. Customers will also be able to delay activation for several uses of the product, until a time that is convenient for them. For those who obtain a copy of the software illegally, Product Activation will make their inappropriate use of the product more difficult. Millions of customers have used Product Activation to date with little or no difficulty.

How will Microsoft Product Activation help thwart piracy?

Product Activation will help reduce casual copying by ensuring that the copy of the software Product Ceing installed is legal and has been installed on a PC in compliance with the End User License Agreement (EULA). Installations beyond those allowed in the license agreement will fail to activate.

Haven't companies tried to implement anti-piracy technologies before and failed?

Anti-piracy technologies that have been used in the past have not been easy for customers to use and were generally viewed as unacceptable by customers and the industry. For example, some early PC products required specialized hardware components or boot diskettes that were cumbersome for the user. Product Activation is a breakthrough technology in that it makes activation a natural part of setting up the software and avoids the pitfalls of anti-piracy methods used in the early days of the PC industry.

How does the customer benefit from this approach?

Over time, reduced piracy means that the software industry can invest more in product development, quality and support. This ensures better products and more innovation for customers. Ultimately, customers will benefit from the economic impact of reduced piracy through more jobs and higher wages. Customers will also receive the best value *for their software investment by being able to receive product updates and other product information. Product Activation also helps prevent unsuspecting customers from purchasing counterfeit software. Customers who purchase counterfeit products could find they are missing key elements, such as user manuals, product keys, certificates of authenticity and even software code. They may also find that the counterfeit software contains viruses or does not work as well as the genuine product does.*

What were some of the key lessons learned by Microsoft from the pilot of activation with Office 2000?

Customers generally found activation to be easy and unobtrusive. Telephone calls average two to three minutes, with hold times of two to three minutes or less. On average, over 70 percent of the activation requests are through the Internet and approximately 2 percent of activation requests are due to hardware changes or other reactivations.

Won't hackers and pirates just crack this like they did earlier copy-protection attempts? Will this really help stop piracy?

Product Activation is not a single "silver bullet" solution to global piracy. It is, however, significantly more sophisticated than past methods and is not easy for would-be pirates to circumvent. At the same time, it is a simple and unobtrusive process for legal customers. It will help prevent casual copying of software, which is by far the most prevalent and damaging type of software piracy. It is not designed to target sophisticated and organized criminal counterfeiters.

Won't Product Activation be easy to crack? I've heard on the Internet that it will be cracked before the product is released. I also heard you can buy Office XP final product on the street in Asia. Doesn't this mean the code has been cracked?

Product Activation has yet to be cracked. The so-called "crack" now being passed around the Internet contains a set of instructions for setting a registry key that disables activation. Microsoft made the existence of this registry key public to its technical beta testers in early February and included it as a testing tool, telling them where it was and how to set it to disable activation. That said, the intellectual property protection arena is a cat-and-mouse game. All IP protection technologies will be cracked at some point; it is just a matter of time. The measure of success is not completely stopping software piracy. Success is more likely measured in increased awareness of the terms of the license agreement and increased license compliance.

Is there rechecking of the activation done after initial activation? Is there any secret data transfer to Microsoft?

The product does check itself from time to time to see if it is activated and if it is still on the same PC on which it was originally activated, but at no time whatsoever is information transferred to Microsoft as a result of Product Activation except while the user is actually in the process of activating the system. There absolutely is no "secret" data transfer.

Will Microsoft use activation to force me to upgrade? In other words, will Microsoft ever stop giving out activation codes for any of the products that require activation?

No, Microsoft will not use activation as a tool to force people to upgrade. Activation is merely an anti-piracy tool, nothing else.

Microsoft will also support the activation of Windows XP throughout its life and will likely provide an update that turns activation off at the end of the product's lifecycle so users would no longer be required to activate the product.

Appendix 2. Our Cryptography Implementation and Its Security Analysis

Mathematical Aspects

System and key parameter generation

```
1) Define NKey = 1024
2) Generating big prime numbers M, P; M < P
3) A random vector is generated SK[i] i=[1..NKey] - a private key
4) Public key PK[i] = M^SK[i] mod P
5) Constants M,P,PK[] are hard-coded into the program
```

Signature (user key)

```
1) H[] = Hash(RegistrationName)
2) K = random
3) R = DHH(M^K + H[0])
4) S = K - (SK[1]*(H[1] xor R) + SK[2]*(H[2] xor R) + ... + SK[NKey]*(H[NKey] xor R))
5) A pair of R,S is transmitted to the user as a key for the RegistrationName.
```

Signature verification

```
1) H[] = Hash(RegistrationName)
2) Y = PK[1]^(H[1] xor R)) * (PK[2]^(H[2] xor R)) * ... * (PK[NKey]^(H[NKey] xor R)
3) Check R == DHH((M^S) * Y + H[0])
```

*) Hash() - hash function, DHH() - very slow hash function (30 op/sec)

*) all arithmetical operations are performed by modulo P

Security Analysis

Public Key Brute Force Attack

A system will be broken in case of recovery of a private key. To recover a private key it is necessary to solve NKey equations M^X mod P = PK relative to X. In case of a brute force attack searching X on a PC like PIII-800 MHz (3*10^3 modular exponent operations per second), we'll get the following table to solve one equation.

SizeOf(X), bits	Break days
31	8
32	16
34	66
36	265
38	10^3
40	4*10^3
45	1,5*10^5
50	4*10^6
55	1,5*10^8
60	4*10^9

Parallel brute force attack on a system of 1024 equations increases required time by 10% per equation. Currently we use 62 bit integers.

Public Key Pollard-rho Attack

Solving the M^X mod P = PK can be significantly accelerated by using the Pollard-rho method. This method allows solving the equation for

`Sqrt(3,14*2^N/2) ~= 1.77*2^((N-1)/2)`

operations (where N - is a width of X in bits, and an 'operation' is modular exponent). Assuming the speed of 3*10^3 operations per second we'll get 31 days for solving equation with SizeOf(X) = 62 bits.

For a public key attack it is necessary to solve NKey equations. The time of breaking the system of NKey equation will be 3*10^4 days.

No method currently exists that allows finding a discrete logarithm for time better than Sqrt(2^N) = 2^(N/2). Note that some faster methods of solving the discrete logarithm problem may require up to 2^(N+3) bytes of memory.

Long Signature Brute Force

The purpose of this attack is to generate a valid signature using brute force.

1) H[] = Hash(RegistrationName)
2) take some R
2) Y = PK[1]^(H[1] xor R)) * (PK[2]^(H[2] xor R)) * ... * (PK[NKey]^(H[NKey] xor R)
3) searching S until R == DHH((M^S) * Y + H[0])

As a DHH hash is calculated very slowly (approximately 30 operations per second on PIII-800), the attack time can be estimated as follows.

SizeOf(R), bits	**Break days**
24	13
26	55
28	220
30	10^3
32	3,5*10^3

In current implementation R is assumed to be 31bits for registration key and 35 bits for activation key.

The foregoing results conclude that attacks weaker than Pollard-rho are pointless. This, however, requires a certain mathematical groundings from the attacker.