

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

Inside Microsoft's Windows Product Activation

Berlin, Germany, July 9, 2001. Fully Licensed GmbH has analyzed the internals of Windows Product Activation, Microsoft's anti-piracy technology built into Windows XP. "We contribute technical facts to a discussion that is currently characterized by uncertainty and speculation about XP revealing details of a user's hardware configuration, the installed software, or even personal data during the activation procedure." says Thomas Lopatic, managing director and CTO of Fully Licensed. The comprehensive technical analysis performed by the licensing experts proves that in addition to insensitive hardware-related information only the individual serial number of each copy of Windows XP is transmitted.

Windows Product Activation is designed to enforce the licensing terms that govern the use of Windows XP by binding an XP license to a single hardware configuration and thus to a single computer. While the activation procedure requires data to be transmitted to an activation center, Microsoft declines to publish any technical details of the information revealed by this data transmission. The lack of facts has given rise to speculation. Some users fear that modifying their hardware configuration would require re-activation of Windows XP. Others fear that details of their hardware configuration could be embedded in the transmitted data.

To satisfy the need for technical details, Fully Licensed GmbH has performed an in-depth analysis of Windows Product Activation as implemented by Windows XP Release Candidate 1. The experts discovered that ten different hardware components form the basis for a hardware ID, which is sent to the activation central during activation. However, due to the method employed to generate the hardware ID, it is very likely that many hardware configurations result in the same ID. Consequently, determining the actual hardware configuration corresponding to a given hardware ID is an infeasible task. In addition to the hardware ID only information derived from the product key - a kind of serial number accompanying each distributed copy of Windows XP - is transmitted.

Moreover, typical hardware updates do not pose any problems either. "More than three of the ten hardware components considered when calculating the hardware ID have to be replaced - e.g. the harddrive, the CD-ROM drive, the microprocessor, and the network adapter - to make re-activation necessary. If the hardware ID is associated with a notebook that supports a docking-station, the policy is still more liberal." says Thomas Lopatic.

"Since our analysis proves that the transmitted information is completely innocuous, we are surprised that Microsoft has been that vague about the inner workings of WPA for all these months." Says Matthias Kunze, managing director and CFO of Fully Licensed. "Software piracy is still a major problem for all software companies. And we think that their interest in raising the bar for software pirates is absolutely justified." he adds.

In addition to relying on technical methods to tackle software piracy Fully Licensed advocate complementary means to form a holistic approach to the problem. Advanced licensing models - e.g. software rental or software leasing - offer a commercially attractive alternative to the use of pirated software in companies. The necessary technological basis has already moved from hype to here. "Software watermarking, automated code obfuscation, online license enforcement, and license management form parts of the Fully Licensed holistic solution for software licensing and software distribution." Says Thomas Lopatic.

The technical paper covering the in-depth analysis of Windows Product Activation as well as a demonstration program including source code is available from the Fully Licensed website at <http://www.licenturion.com/xp/>.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

About Fully Licensed GmbH

Fully Licensed GmbH provides a secure and flexible infrastructure for online licensing and online distribution of software. It integrates software watermarking, automated code obfuscation, license enforcement and license management, while preserving the end-user's right to privacy. In addition to raising the bar for software pirates by technical means, this approach supplies the basis for easy and secure realization of advanced licensing models such as software rental or software leasing. Fully Licensed GmbH consider themselves to be an independent and unbiased mediator between software vendors and end-users. Their research and development branch every now and then analyzes licensing solutions implemented by other companies.

INTRODUCTION

The current public discussion of Windows Product Activation (WPA) is characterized by uncertainty and speculation. In this paper we supply the technical details of WPA - as implemented in Windows XP – that Microsoft should have published long ago.

While we strongly believe that every software vendor has the right to enforce the licensing terms governing the use of a piece of licensed software by technical means, we also do believe that each individual has the right to detailed knowledge about the full implications of the employed means and possible limitations imposed by it on software usage.

In this paper we answer what we think are currently the two most important open questions related to Windows Product Activation.

- Exactly what information is transmitted during activation?
- How do hardware modifications affect an already activated installation of Windows XP?

Our answers to these questions are based on Windows XP Release Candidate 1 (build 2505). Later builds as well as the final version of Windows XP might differ from build 2505, e.g. in the employed cryptographic keys or the layout of some of the data structures.

However, beyond such minor modifications we expect Microsoft to cling to the general architecture of their activation mechanism. Thus, we are convinced that the answers provided by this paper will still be useful when the final version of Windows XP ships.

This paper supplies in-depth technical information about the inner workings of WPA. Still, the discussion is a little vague at some points in order not to facilitate the task of an attacker attempting to circumvent the license enforcement supplied by the activation mechanism.

XPDec, a command line utility suitable for verifying the presented information, can be obtained from <http://www.licenturion.com/xp/>. It implements the algorithms presented in this paper. Reading its source code, which is available from the same location, is highly recommended.

We have removed an important cryptographic key from the XPDec source code. Recompiling the source code will thus fail to produce a working executable. The XPDec executable on our website, however, contains this key and is fully functional.

So, download the source code to learn about the inner workings of WPA, but obtain the executable to experiment with your installation of Windows XP.

We expect the reader to be familiar with the general procedure of Windows Product Activation.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

INSIDE THE INSTALLATION ID

We focused our research on product activation via telephone. We did so, because we expected this variant of activation to be the most straight-forward to analyze.

The first step in activating Windows XP via telephone is supplying the call-center agent with the Installation ID displayed by msoobe.exe, the application that guides a user through the activation process. The Installation ID is a number consisting of 50 decimal digits that are divided into groups of six digits each, as in

```
002666-077894-484890-114573-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XX
```

In this authentic Installation ID we have substituted digits that we prefer not to disclose by 'X' characters.

If msoobe.exe is invoked more than once, it provides a different Installation ID each time.

In return, the call-center agent provides a Confirmation ID matching the given Installation ID. Entering the Confirmation ID completes the activation process.

Since the Installation ID is the only piece of information revealed during activation, the above question concerning the information transmitted during the activation process is equivalent to the question

```
'How is the Installation ID generated?'
```

To find an answer to this question, we trace back each digit of the Installation ID to its origins.

Check digits

The rightmost digit in each of the groups is a check digit to guard against simple errors such as the call center agent's mistyping of one of the digits read to him or her. The value of the check digit is calculated by adding the other five digits in the group, adding the digits at even positions a second time, and dividing the sum by seven. The remainder of the division is the value of the check digit. In the above example the check digit for the first group (6) is calculated as follows.

```
  1 | 2 | 3 | 4 | 5  <- position
  ---+---+---+---+---
  0 | 0 | 2 | 6 | 6  <- digits

  0 + 0 + 2 + 6 + 6 = 14      (step 1: add all digits)
    0   + 6   + 14 = 20     (step 2: add even digits again)

step 3: division
      20 / 7 = 2, remainder is 20 - (2 * 7) = 6

=> check digit is 6
```

Adding the even digits twice is probably intended to guard against the relatively frequent error of accidentally swapping two digits while typing, as in 00626 vs. 00266, which yield different check digits.

Decoding

Removing the check digits results in a 41-digit decimal number. A decimal number of this length roughly corresponds to a 136-bit binary number. In fact, the 41-digit number is just the decimal

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

encoding of such a 136-bit multi-precision integer, which is stored in little endian byte order as a byte array. Hence, the above Installation ID can also be represented as a sequence of 17 bytes as in

```
0xXX 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX 0xXX
0x94 0xAA 0x46 0xD6 0x0F 0xBD 0x2C 0xC8
0x00
```

In this representation of the above Installation ID 'X' characters again substitute the digits that we prefer not to disclose. The '0x' prefix denotes hex notation throughout this paper.

Decryption

When decoding arbitrary Installation IDs it can be noticed that the most significant byte always seems to be 0x00 or 0x01, whereas the other bytes look random. The reason for this is that the lower 16 bytes of the Installation ID are encrypted, whereas the most significant byte is kept in plaintext.

The cryptographic algorithm employed to encrypt the Installation ID is a proprietary four-round Feistel cipher. Since the block of input bytes passed to a Feistel cipher is divided into two blocks of equal size, this class of ciphers is typically applied to input blocks consisting of an even number of bytes - in this case the lower 16 of the 17 input bytes. The round function of the cipher is the SHA-1 message digest algorithm keyed with a four-byte sequence.

Let + denote the concatenation of two byte sequences, ^ the XOR operation, L and R the left and right eight-byte input half for one round, L' and R' the output halves of said round, and First-8() a function that returns the first eight bytes of an SHA-1 message digest. Then one round of decryption looks as follows.

```
L' = R ^ First-8(SHA-1(L + Key))
R' = L
```

The result of the decryption is 16 bytes of plaintext, which are - together with the 17th unencrypted byte - from now on interpreted as four double words in little endian byte order followed by a single byte as in

name	size	offset
H1	double word	0
H2	double word	4
P1	double word	8
P2	double word	12
P3	byte	16

H1 and H2 specify the hardware configuration that the Installation ID is linked to. P1 and P2 as well as the remaining byte P3 contain the Product ID associated with the Installation ID.

Product ID

The Product ID consists of five groups of decimal digits, as in

```
AAAAA-BBB-CCCCCC-DDEEE
```

If you search your registry for a value named 'ProductID', you will discover the ID that applies to your installation. The 'About' window of Internet Explorer should also yield your Product ID.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

Decoding

The mapping between the Product ID in decimal representation and its binary encoding in the double words P1 and P2 and the byte P3 is summarized in the following table.

digits	length	encoding
AAAAA	17 bits	bit 0 to bit 16 of P1
BBB	10 bits	bit 17 to bit 26 of P1
CCCCCC	28 bits	bit 27 to bit 31 of P1 (lower 5 bits) bit 0 to bit 22 of P2 (upper 23 bits)
DDEEE	17 bits	bit 23 to bit 31 of P2 (lower 9 bits) bit 0 to bit 7 of P3 (upper 8 bits)

The meaning of each of the five groups of digits is documented in the next table.

digits	meaning
AAAAA	apparently always 55034 (in Windows XP RC1)
BBB	most significant three digits of Raw Product Key (see below)
CCCCCC	least significant six digits of Raw Product Key plus check digit (see below)
DD	index of the public key used to verify the Product Key (see below)
EEE	random value

As can be seen, the (Raw) Product Key plays an important role in generating the Product ID.

Product Key

The Raw Product Key is buried inside the Product Key that is printed on the sticker distributed with each Windows XP CD. It consists of five alphanumeric strings separated by '-' characters, where each string is composed of five characters, as in

FFFFF-GGGGG-HHHHH-JJJJJ-KKKKK

Each character is one of the following 24 letters and digits:

B C D F G H J K M P Q R T V W X Y 2 3 4 6 7 8 9

Very similar to the decimal encoding of the Installation ID the 25 characters of the Product Key form a base-24 encoding of the binary representation of the Product Key. Decoding the Product Key yields a multi-precision integer of roughly 115 bits, which is stored – again in little endian byte order - in an array of 15 bytes. Decoding the above Product Key results in the following byte sequence.

0x6F 0xFA 0x95 0x45 0xFC 0x75 0xB5 0x52
0xBB 0xEF 0xB1 0x17 0xDA 0xCD 0x00

Of these 15 bytes the least significant four bytes contain the Raw Product Key in little endian byte order. The least significant bit is removed by shifting this 32-bit value (0x4595FA6F - remember the little endian byte order) to the left by one bit position, resulting in a Raw Product Key of 0x22CAFD37, or

583728439

in decimal notation.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

The eleven remaining bytes form a digital signature, allowing verification of the authenticity of the Product Key by means of a hard-coded public key.

Product Key -> Product ID

The three most significant digits, i.e. 583, of the Raw Product Key's nine-digit decimal representation directly map to the BBB component of the Product ID described above.

To obtain the CCCCCC component, a check digit is appended to the remaining six digits 728439. The check digit is chosen such that the sum of all digits - including the check digit - is divisible by seven. In the given case, the sum of the six digits is

$$7 + 2 + 8 + 4 + 3 + 9 = 33$$

which results in a check digit of 2, since

$$7 + 2 + 8 + 4 + 3 + 9 + 2 = 33 + 2 = 35$$

which is divisible by seven. The CCCCCC component of the Product ID is therefore 7284392.

For verifying a Product Key, more than one public key is available. If verification with the first public key fails, the second is tried, etc. The DD component of the Product ID specifies which of the public keys in this sequence was successfully used to verify the Product Key.

This mechanism might be intended to support several different parties generating valid Product Keys with different individual private keys.

However, the different private keys might also represent different versions of a product. A Product Key for the 'professional' release could then be signed with a different key than a Product Key for the 'server' release. The DD component would then represent the product version.

Finally, a valid Product ID derived from our example Product Key might be

55034-583-7284392-00123

which indicates that the first public key (DD = index = 0) matched and 123 was chosen as the random number EEE.

The randomly selected EEE component is the reason for msoobe.exe presenting a different Installation ID at each invocation. Because of the applied encryption this small change results in a completely different Installation ID.

So, the Product ID transmitted during activation will most probably differ in the last three digits from your Product ID as displayed by Internet Explorer or as stored in the registry.

Hardware Information

As discussed above, the hardware configuration linked to the Installation ID is represented by the two double words H1 and H2.

Bit-fields

For this purpose, the double words are divided into twelve bit-fields. The relationship between the computer hardware and the bit-fields is given in the following table.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

double word	offset	length	bit-field value based on
H1	0	10	volume serial number string of system volume
H1	10	10	network adapter MAC address string
H1	20	7	CD-ROM drive hardware identification string
H1	27	5	graphics adapter hardware identification string
H2	0	3	unused, set to 001
H2	3	6	CPU serial number string
H2	9	7	harddrive hardware identification string
H2	16	5	SCSI host adapter hardware identification string
H2	21	4	IDE controller hardware identification string
H2	25	3	processor model string
H2	28	3	RAM size
H2	31	1	1 = dockable 0 = not dockable

Bit 31 of H2 specifies, whether the bit-fields represent a notebook computer that supports a docking station. If docking is possible, the activation mechanism will be more tolerant with respect to future hardware modifications. Here, the idea is that plugging a notebook into its docking station possibly results in changes to its hardware configuration, e.g. a SCSI host adapter built into the docking station may become available.

Bits 2 through 0 of H2 are unused and always set to 001.

If the hardware component corresponding to one of the remaining ten bit-fields is present, the respective bit-field contains a non-zero value describing the component. A value of zero marks the hardware component as not present.

All hardware components are identified by a hardware identification string obtained from the registry. Hashing this string provides the value for the corresponding bit-field.

Hashing

The hash result is obtained by feeding the hardware identification string into the MD5 message digest algorithm and picking the number of bits required for a bit-field from predetermined locations in the resulting message digest. Different predetermined locations are used for different bit-fields. In addition, a hash result of zero is avoided by calculating

$$\text{Hash} = (\text{Hash} \% \text{BitFieldMax}) + 1$$

where BitFieldMax is the maximal value that may be stored in the bit-field in question, e.g. 1023 for a 10-bit bit-field, and 'x % y' denotes the remainder of the division of x by y. This results in values between 1 and BitFieldMax. The obtained value is then stored in the respective bit-field.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

RAM bit-field

The bit-field related to the amount of RAM available to the operating system is calculated differently. The seven valid values specify the approximate amount of available RAM as documented in the following table.

value	amount of RAM available
0	(bit-field unused)
1	below 32 MB
2	between 32 MB and 63 MB
3	between 64 MB and 127 MB
4	between 128 MB and 255 MB
5	between 256 MB and 511 MB
6	between 512 MB and 1023 MB
7	above 1023 MB

It is important to note that the amount of RAM is retrieved by calling the GlobalMemoryStatus() function, which reports a few hundred kilobytes less than the amount of RAM physically installed. So, 128 MB of RAM would typically be classified as "between 64 MB and 127 MB".

Real-world example

Let us have a look at a real-world example. On one of our test systems the hardware information consists of the following eight bytes.

```
0xC5 0x95 0x12 0xAC 0x01 0x6E 0x2C 0x32
```

Converting the bytes into H1 and H2, we obtain

```
H1 = 0xAC1295C5 and H2 = 0x322C6E01
```

Splitting H1 and H2 yields the next table in which we give the value of each of the bit-fields and the information from which each value is derived.

dw & offset	value	derived from
H1 0	0x1C5	'1234-ABCD'
H1 10	0x0A5	'00C0DF089E44'
H1 20	0x37	'SCSI\CDROMPLEXTOR_CD-ROM_PX-32TS__1.01'
H1 27	0x15	'PCI\VEN_102B&DEV_0519&SUBSYS_00000000&REV_01'
H2 0	0x1	(unused, always 0x1)
H2 3	0x00	(CPU serial number not present)
H2 9	0x37	'SCSI\DISKIBM_____DCAS-34330_____S65A'
H2 16	0x0C	'PCI\VEN_9004&DEV_7178&SUBSYS_00000000&REV_03'
H2 21	0x1	'PCI\VEN_8086&DEV_7111&SUBSYS_00000000&REV_01'
H2 25	0x1	'GenuineIntel Family 6 Model 3'
H2 28	0x3	(system has 128 MB of RAM)
H2 31	0x0	(system is not dockable)

Using XPDec

XPDec is a utility to be run from the command prompt. It may be invoked with one of four command line options to carry out one of four tasks.

XPDec -i

This option enables you to access the information hidden in an Installation ID. It decodes the Installation ID, decrypts it, and displays the values of the hardware bit-fields as well as the Product ID

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

of your product. Keep in mind that the last three digits of the Product ID contained in the Installation ID are randomly selected and differ from the Product ID displayed by Internet Explorer. The only argument needed for the '-i' option is the Installation ID, as in

```
XPDec -i 002666-077894-484890-114573-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XX
```

XPDec -p

To help you trace the origin of your Product ID, this option decodes a Product Key and displays the Raw Product Key as it would be used in a Product ID. The only argument needed for the '-p' option is the Product Key, as in

```
XPDec -p FFFFFF-GGGGGG-HHHHHH-JJJJJJ-KKKKKK
```

Note that this option does not verify the digital signature of the Product Key.

XPDec -v

This option calculates the hash of a given volume serial number. It was implemented to illustrate our description of string hashing. First use '-i' to display the hardware bit-fields. Then use this option to verify our claims concerning the volume serial number hash. The only argument needed for the '-v' option is the volume serial number of your system volume, as in

```
XPDec -v 1234-ABCD
```

(The volume serial number is part of the 'dir' command's output.)

XPDec -m

This option calculates the network adapter bit-field value corresponding to the given MAC address. Similar to '-v' this option was implemented as a proof of concept. The only argument needed for the '-m' option is the MAC address of your network adapter, as in

```
XPDec -m 00-C0-DF-08-9E-44
```

(Use the 'route print' command to obtain the MAC address of your network adapter.)

HARDWARE MODIFICATIONS

When looking at the effects of hardware modifications on an already activated installation of Windows XP, the file 'wpa.dbf' in the 'system32' directory plays a central role. It is a simple RC4-encrypted database that stores, among other things like expiration information and the Confirmation ID of an activated installation,

- the bit-field values representing the current hardware configuration (a), and
- the bit-field values representing the hardware configuration at the time of product activation (b).

While a) is automatically updated each time the hardware configuration is modified in order to reflect the changes, b) remains fixed. Hence, b) can be thought of as a snapshot of the hardware configuration at the time of product activation.

This snapshot does not exist in the database before product activation and if we compare the size of 'wpa.dbf' before and after activation, we will notice an increased file size. This is because the snapshot is added to the database.

Inside Windows Product Activation

A Fully Licensed Paper

Fully Licensed GmbH, Rudower Chaussee 29, 12489 Berlin, Germany

When judging whether re-activation is necessary, the bit-field values of a) are compared to the bit-field values of b), i.e. the current hardware configuration is compared to the hardware configuration at the time of activation.

Non-dockable computer

Typically all bit-fields with the exception of the unused field and the 'dockable' field are compared. If more than three of these ten bit-fields have changed in a) since product activation, re-activation is required. This means, for example, that in our above real-world example, we could replace the harddrive and the CD-ROM drive and substantially upgrade our RAM without having to re-activate our Windows XP installation. However, if we completely re-installed Windows XP, the information in b) would be lost and we would have to re-activate our installation, even if we had not changed our hardware.

Dockable computer

If bit 31 of H2 indicates that our computer supports a docking station, however, only seven of the ten bit-fields mentioned above are compared. The bit-fields corresponding to the SCSI host adapter, the IDE controller, and the graphics board are omitted. But again, of these remaining seven bit-fields, only up to three may change without requiring re-activation.

CONCLUSIONS

In this paper we have given a technical overview of Windows Product Activation as implemented in Windows XP. We have shown what information the data transmitted during product activation is derived from and how hardware upgrades affect an already activated installation.

Looking at the technical details of WPA, we do not think that it is as problematic as many people have expected. We think so, because WPA is tolerant with respect to hardware modifications. In addition, it is likely that more than one hardware component map to a certain value for a given bit-field. From the above real-world example we know that the PX-32TS maps to the value $0x37 = 55$. But there are probably many other CD-ROM drives that map to the same value. Hence, it is impossible to tell from the bit-field value whether it is a PX-32TS that we are using or one of the other drives that map to the same value.

In contrast to many critics of Windows Product Activation, we think that WPA does not prevent typical hardware modifications and, moreover, respects the user's right to privacy.

ABOUT THE AUTHORS

Fully Licensed GmbH is a start-up company focusing on novel approaches to online software licensing and distribution. Have a look at their website at <http://www.licenturion.com> for more information. Their research branch every now and then analyzes licensing solutions implemented by other companies.

COPYRIGHT

Copyright (C) 2001 Fully Licensed GmbH (www.licenturion.com) All rights reserved.

You are free to do whatever you want with this paper. However, you have to supply the URL of its online version <http://www.licenturion.com/xp/> with any work derived from this paper to give credit to its authors.