


Software Activation and Registration via Web Application

KULHÁNEK, Jiří¹ & TŮMA, Jiří²

¹ Ing., Ph.D., ✉ Katedra ATR-352, VŠB-TU Ostrava, 17. listopadu, Ostrava - Poruba, 708 33  jiri.kulhanek@vsb.cz

² prof., Ing., Ph.D. ✉ Katedra ATR-352, VŠB-TU Ostrava, 17. listopadu, Ostrava - Poruba, 708 33  jiri.tuma@vsb.cz

Abstract: *The last stage of software development is to sell or distribute application to customer. Typically, the software developer wants to limit unauthorized reselling or copying of product, or wants to know who is using his product. This can be done by many different methods, but in last time the interesting method is online software activation and online software registration.*

The OS Windows (and MS Office etc.) software implement theirs own sophisticated methods to do activation and registration, unfortunately this method is possible to use with this product only.

Indeed, we have to develop some own online activation system to do this and implement activation client into our software.

Keywords: *software activation, security, distributed*

1 Activation of applications

At first definition of product activation from [wikipedia]

“Product activation is a license validation procedure required by some computer software programs. Specifically, product activation refers to a method where a software application hashes hardware serial numbers and an ID number specific to the product's license (a product key) to generate a unique Installation ID. This Installation ID is sent to the manufacturer to verify the authenticity of the product key and to ensure that the product key is not being used for multiple installations.”

As mentioned in definition, the activation consists of several steps:

- To create an registration code and distribute this code to user
- To compute some installation hash code and send this code to activation centre
- To validate received hash code, dig the integrated registration code and send activation acceptance or rejection to user (by another hash code)

Well known product activation example is Microsoft Windows XP, Vista, Office 2003 and similar products.

Why to use product activation?

- To avoid software piracy
- To increase sales (as result of avoiding software piracy)
- To create connection between customers and software creators
- To distinguish between free and licensed version and lock/unlock features

2 Product activation in MS Windows

As mentioned in chapter 1, the most known product activation process is activation of MS Windows OS. I'll shortly summary the Microsoft activation mechanism.

- during the windows product installation the user have to type some valid product key, the validity of product key is checked by some quite easy algorithm
- after installation, the user is warned to activate and the 30days activation period is started
- after 30days activation period expire, the software is disabled
- the activation can be done online by internet connection or by phone (call to "green" line)

3 Own product activation mechanism

We have to write our own activation mechanism, so we have to define why we want to use activation. We are planning to integrate activation mechanism with our medium sized software for signal measuring and processing. This is not typical end user software and we have these priorities:

- to know how many copies of our product is in use
- to try to communicate with customers of our product
- to give some additional features to customers

We don't want to eliminate software privacy (for now), but only to watch using of software. Our goals are follows:

- to ensure that each licensed user have to type some product key
- to ensure that each product key has to be activated by internet
- to count number of product activation and to count multiple activations with the same product key

These goals have to be achieved by distributed heterogeneous application, the application consist of:

- Database with generated product keys (customer names or some sales information) and count of activations of this number
- Web pages for automated or manual product activation
- Web pages for activation database administration
- In product activation mechanism.

Appliacation layer have to:

- Generate new product keys to database on request
- Validate product key in product installation
- Generate successful activation ID

The process of generating product keys has to:

- ensure some nontrivial keys, which can't be simulated and generated by common users
- ensure that each key can be generate only once
- ensure some simply check algorithm of key validity

3 Product keys generating

There are many methods to generate unique keys; each of them is based on very long numbers. The first task to do is to develop some basic mathematic methods (sum, multiply,

pow etc.) with very long numbers. The product keys are typically written in 36char alphabet and their length is typically 16 positions (for example X534-TYW2-441F-LWF3). Such number is has 35^{16} combinations – it is approximately 2^{82} combinations. For such computation we need more than 64 bit numbers, but common today's CPU's are able to do internal mathematic operations up to 64bits.

This can be for plus and minus operation realized quite easy by sequenced plus or minus operations with one bit shift to higher sequence. But the multiply algorithm is more complicated. Because of that we need some mathematic algorithm to computation with variable length numbers, we need plus, minus, multiply and divide and modulo operation. Unfortunately, common programming languages didn't have any support for such computations, so we have to develop our own algorithms. The good information is, that product key generation and validation will be deterministic operation with only a few of such long number operations, so we can create robust and slow algorithm without affecting the performance.

The product key generating have to ensure nontrivial key generating, the easy way to do this is to base keys on some non public key template – for example : KEYT-EMPL-ATEF-ORME. Then we have to such key template change for each generated key. The usable basic operation is multiply of key template with some prime number. In such way we ensure massively change of generated keys between sequent keys and ensure unique keys too (because of prime numbers).

For realization of product keys generation and validation we have to create this mathematic algorithm with variant length integers:

- sum
- odds
- multiply
- division
- modulo

Then we have to convert our variant length binary integers from/to specified alphabet (for example alphabet of 35).

Product key generation:

Key = KeyTemplate * unique_primenumber

Product key validation

Key modulo KeyTemplate have to be 0

Key / KeyTemplate have to be a prime number.

How to generate sequel of unique prime numbers? The easy way is to search for prime numbers from some starting number and this starting point move after each successfully founded prime number. For speed reasons the prime numbers should be integers.

4 Product key hashing

After a user type product key to installed software the simple validation occurs (chapter 3). Then we have to do product activation via internet. Internet validation can be done by online automatic or offline manual way. In automatic way the installed application communicates hiddenly with activation server and everything is ok. In case of computer not connected to internet we have to ensure activation in some manual offline way.

Typically the computer generates some key and user sends this key to activation server (by any internet browser), then user receives product key response and types it to installed computer. It is potentially dangerous, because the user get question to and answer from activation mechanism – now the user can repeatedly install and activate products without communicate with activation server. To avoid this behaviour the key generated for activation

is expanded of some additional information (typically the processor ID and time stamp, mac address or something else). The activation server separate this additional information from received key, check the key against key database and if it's ok then generate good response.

The activation server response is expanded by information received from source computer, by this way the sended data and received data are valid only for specific computer and specific time moment (for example for an one day from activation request generation).

The hashing algorithm can't be trivial – because than user will be able to distinguish product key additional information in activation request and then in activation answer.

Then – expanding of product id by additional information has to be reversible algorithm independent on key or additional data.

$$\text{Hashed_key} = \text{Installation_Key} + \{\text{hardware_id} + \text{creation_time}\}$$

5 Activation response

After client program sends activation request with hashed key the activation server have to decompose the hashed key to original installation key and additional hash sequence, then the installation key is searched in key database and if the key is found then the positive response is created.

The question is, how the client program distinguishes between positive and false response.

Again the response creation has to be nontrivial but deterministic. The only secure method is to use some cryptographic method to encode response. In our activation system we use only simple method of sending key generated with unique prime number +1. In such manner the response is not unique, but is quite simply and enough for our purposes.

$$\text{Response Key} = \text{Template key} * (\text{unique_primenumber} + 1)$$

The client computer only again check validity of key against primenumber+1. The security of algorithm is done by non public template key.

6 Conclusions

The problematic of product activation is realized by several commercial systems, which can be purchased and integrated with an application. This paper shortly describes very simple method of activation based of non-public template-key.

This method is can be extended by principles of public keys, then response of activation system will be the second (private) part of key. Such system is very complicated to hack, but quite to difficult to implement too. The research work was performed to financial support of grant reg. No 101/07/1345 of GA CZ.

7 References

BABIUCH, M. Vývoj webových aplikací pro podporu výuky v laboratořích automatického řízení Journal of Cybernetics and Informatics September 2004, Special Issue "New Trends in Education of Automation and Information Technology 2004. ISSN: 1336-4774

WIKIPEIDA, *Product activation*. [online], [cit. 26.4.2007] Available from www: <URL: http://en.wikipedia.org/wiki/Product_activation>