A PKI Case Study on the Issues of the Personal Security Profile for Web and Mail Agents

Marius Marian*, Andrei Pîrvan**

**Department of Computer Science and Information Technology, University of Craiova, RO 200440, Craiova, Romania (Tel: +40-251-438.198; e-mail: marius.marian@cs.ucv.ro).*
***University of Craiova, Romania, RO 200440 (e-mail:andrei.pirvan@yahoo.com)*

Abstract: These paper details the actual problems encountered while using digital certificates with actual web browsers and mail user agents. After setting up a public-key infrastructure for the community of users within the University of Craiova, we have begun testing the issued digital certificates on different usage scenarios. The personal security profile of a subscriber consists typically of its private key, the corresponding digital certificate, and the chain of hierarchically-related certification authorities' certificates. We have focused on the preliminary step of installing public-key certificates within the secure-credentials repositories of today's most commonly used browsers. The test cases were devised for some of the most popular operating systems employed by the users of our community.

Keywords: public-key infrastructures (PKI), digital certificates, security managers, web browsers.

1. INTRODUCTION

Nowadays when computers are a part of our everyday life, the field of information security has attracted interest of many people and organizations, and its practical application can be observed everywhere. Some of the most common features offered by information security are confidentiality, integrity, availability and non-repudiation of data. One way to achieve them is by using the digital certificates issued by a public key infrastructure.

Such an infrastructure covers all the hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. Essentially, the role of PKIs is to enable trust within end entities employing asymmetric cryptography. Trust is achieved by means of a verifiable digital signature performed by a trusted third party (TTP) on the binding between a public key value and the identity of the entity controlling the corresponding private key. This TTP is also named certification authority (CA) since it guarantees the fact that a public key truly belongs to a particular subscriber of the PKI.

The most common structure of PKIs is hierarchical in which CAs are subordinated to each other in order to delegate the task of issuing certificates to particular domains of related clients. The root CA signs the public keys of only the top-level CAs, these in turn sign the public keys of subordinated CAs and their subscribers, and the process continues down to the last level of the hierarchy.

A PKI end entity must first obtain a pair of keys (one public and one private), and then install its CA-issued digital certificate in the particular application or system in order to benefit of the above mentioned advantages of public-key cryptography. The personal security profile of a PKI subscriber is the set of the public-key certificate, the corresponding private key, and all the certificates of the intermediate CAs in the hierarchy up to the root CA.

2. STATEMENT OF THE PROBLEM

The installation of the personal security profile is neither easy nor always correct. It strictly depends on the browser and the underlying operating system. If PKI is to be widely used, the issues concerning the proper installation of top-level certificates (root CA in first place) must be handled by the PKI team. Users are neither to be charged with extra amount of work or skills nor having them to manage complicated tools for their working IT environment.

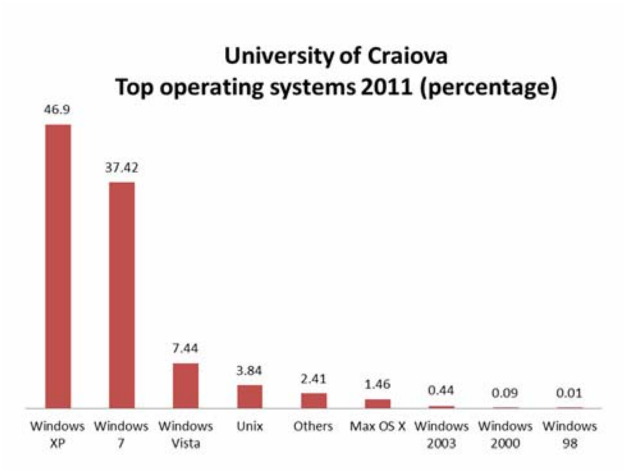


Figure 1. Top operating systems used by the community of University of Craiova (Source: trafic.ro)

We have identified that browsers and operating systems do not handle properly X.509 end-user certificates and root CA certificates.

2.1 Current status

For the IT environment of University of Craiova, we have designed and deployed a public-key infrastructure called the Romanian Certification Authority, see ROCA (2011). The PKI is based on an open-source project called OpenCA. More details are given in Marian (2011). Members of the community can subscribe to the PKI's certification authorities in order to receive a digital certificate. The main purpose of the PKI is to enable and spread the use of security services for web and e-mail agents (i.e. MUA – mail user agent).

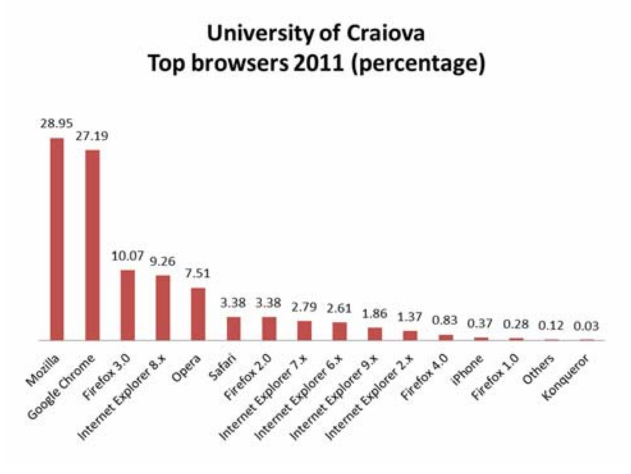
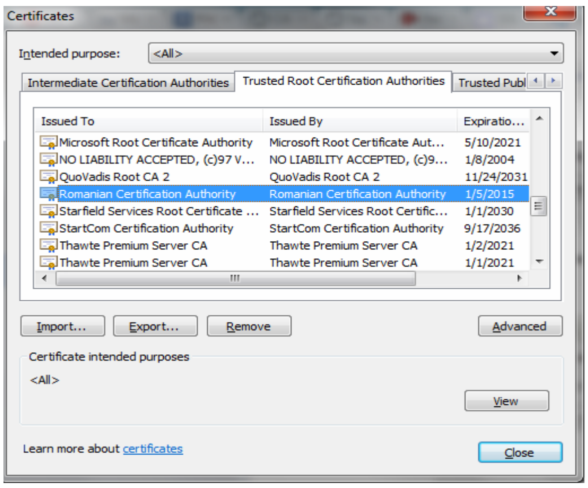


Figure 2. Top browsers used by the UCV community (Source: trafic.ro)

In order to perform the tests with the certificates issued by our PKI, we started to investigate what are the most common combinations of browsers/MUAs and operating systems that are specific to our user community.


Certificates
Intended purpose:
<All>
Intermediate Certification Authorities
Trusted Root Certification Authorities
Trusted Publ
Import...
Export...
Remove
Advanced
Certificate intended purposes
<All>
View
Learn more about certificates
Close

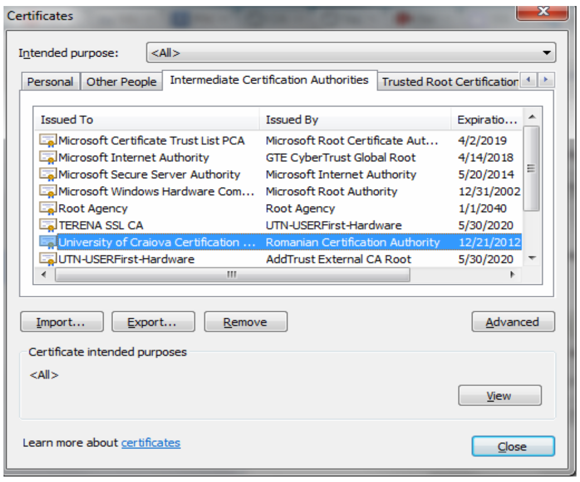
Issued To	Issued By	Expiratio...
Microsoft Root Certificate Authority	Microsoft Root Certificate Aut...	5/10/2021
NO LIABILITY ACCEPTED, (c)97 V...	NO LIABILITY ACCEPTED, (c)9...	1/8/2004
QuoVadis Root CA 2	QuoVadis Root CA 2	11/24/2031
Romanian Certification Authority	Romanian Certification Authority	1/5/2015
Starfield Services Root Certificate ...	Starfield Services Root Certific...	1/1/2030
StartCom Certification Authority	StartCom Certification Authority	9/17/2036
Thawte Premium Server CA	Thawte Premium Server CA	1/2/2021
Thawte Premium Server CA	Thawte Premium Server CA	1/1/2021

Figure 3. Trusted Root Certification Authorities section of the browser repository

Two figures are worth mentioning here. We have conducted an investigation in order to produce as accurate data as possible concerning the utilization of operating systems and browsers within the community of University of Craiova. The results of this investigation (depicted in **Figure 1**, **Figure 2**) are based on the observed network traffic, and on the web statistics provided by a commercial traffic evaluator. The period of the survey was the last trimester of 2011. In what concerns operating systems, we have found out that Microsoft products dominate the community IT environment with a staggering percent of 92.3. On the second place came Unix/Linux with a mere 3.84%. Mobile users account most probably for all the 1.46% traffic originating from Mac OS.

In what concerns the browsers used we have found out that Mozilla products are the most popular ones (accounting for 43.51% of the user community), followed by Google Chrome (27.19%), Microsoft Internet Explorer (16.52%), Opera (7.51%) and Safari (3.75%). We have partially assumed that these figures will also be approximately the same for the top five positions in what concerns the MUAs used within the user community. However it is worth mentioning another interesting behaviour of the user community or at least of a part of it which prefers to switch between proper MUAs and the web-mailing system (web-mail available through the browsers mentioned above).

Also important for the investigation is the fact that some of the browsers are able to store the certificates and corresponding private keys within their own repositories while others rely on the operating system for doing that.

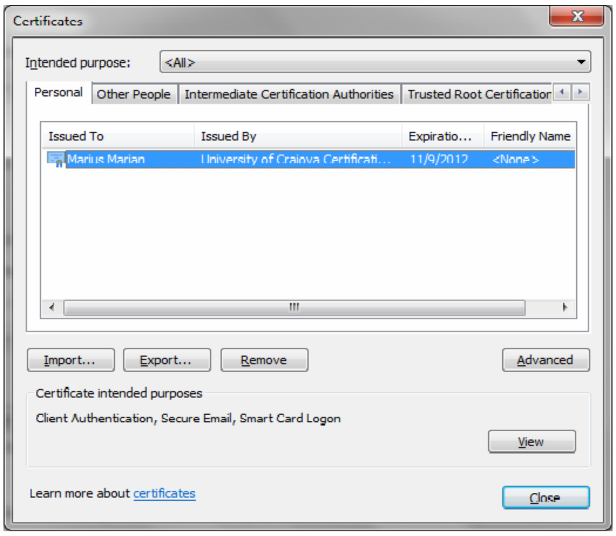

Certificates
Intended purpose:
<All>
Personal
Other People
Intermediate Certification Authorities
Trusted Root Certification
Import...
Export...
Remove
Advanced
Certificate intended purposes
<All>
View
Learn more about certificates
Close

Issued To	Issued By	Expiratio...
Microsoft Certificate Trust List PCA	Microsoft Root Certificate Aut...	4/2/2019
Microsoft Internet Authority	GTE CyberTrust Global Root	4/14/2018
Microsoft Secure Server Authority	Microsoft Internet Authority	5/20/2014
Microsoft Windows Hardware Com...	Microsoft Root Authority	12/31/2002
Root Agency	Root Agency	1/1/2040
TERENA SSL CA	UTN-USERFirst-Hardware	5/30/2020
University of Craiova Certification ...	Romanian Certification Authority	12/21/2012
UTN-USERFirst-Hardware	AddTrust External CA Root	5/30/2020

Figure 4. Intermediate Certification Authorities section of the browser repository

Based on these findings we have planned and proceeded with the actual testing of generating and installing the certificates on combinations of operating systems and browsers.

The main test scenario assumes the user will use her IT environment (the combination of operating system and browser) to connect via HTTP to the CA's front end (a webserver). Here, she will complete a subscription form requesting thus a digital certificate. This form implies generating the cryptographic pair of keys followed by the verifications performed by the CA. If all verifications are successful, the CA will issue a certificate for the new subscriber, and consequently the CA will notify the user via e-mail about the issuance of the new certificate. The user may follow the hyperlinks in the e-mail message in order to install her certificate on the local browser (or MUA) or into her local operating system repository. An alternative to the e-mail hyperlinks (for the end user) is to employ her local browser (the same with which she completed the subscription form above) to connect to the web repository of the PKI, and download directly from there her newly generated certificate. To end the installation process, the end user must assure herself that the root ROCA certificate and the intermediate CA certificates are also installed in the proper sections of her certificate repository. The root CA certificate of the ROCA PKI must end in the *Trusted Root Certification Authorities* section of the repository (see **Figure 3**), while the intermediate CA certificates must be placed into the *Intermediate Certification Authorities* section (see **Figure 4**). Obviously, the certificate of the end-user will end into the *Personal Certificates* section of the repository (see **Figure 5**).



Issued To	Issued By	Expiratio...	Friendly Name
Marius Marian	University of Craiova Certificati...	11/9/2012	<None>

Certificate intended purposes

Client Authentication, Secure Email, Smart Card Logon

Figure 5. Personal Certificate section of the browser repository

What we have found out is the root CA certificate is not always correctly installed, while the other two types of certificates (intermediate CA and personal) install properly in most cases.

3. INVESTIGATED TECHNOLOGIES

3.1 Microsoft CryptoAPI

The Cryptographic Application Programming Interface (API) is an application programming interface included with every Microsoft Windows operating systems since Windows NT 4.0. It provides cryptographic services to developers trying to secure Windows-based applications (including here Internet Explorer, Outlook Express, Office Outlook, and so forth). This crypto API supports both public-key and symmetric key cryptography.

Microsoft CryptoAPI works with a number of CSPs (Cryptographic Service Provider) that may be installed on a specific machine. CSPs are the modules that do the actual work of encoding and decoding data by performing the cryptographic functions. Vendors of hardware security modules may supply a CSP which works with their hardware.

3.2 Mozilla Open-Source PKI Projects

Mozilla organization makes available two open-source projects for PKI-related support operations for its line of communication products (Firefox browser, Thunderbird MUA, etc.), and also for some other third-party products (i.e. AOL, AIM, OpenOffice, RedHat, Apache, Sun Java, Evolution, Gaim, etc.). These two open-source projects are detailed below.

First, the Network Security Services (NSS) represent an open-source set of libraries designed to support cross-platform development of security-enabled client and server applications. Applications based on NSS support X.509v3 certificates, related public-key cryptographic standards (PKCS#5, PKCS#7, PKCS#11, PKCS#12), SSL/TLS, secure multipurpose Internet extensions (S/MIME) and many other standards.

Second, the Personal Security Manager (PSM) consists of a set of libraries performing cryptographic operations on behalf of a client application. These operations include setting up an SSL/TLS connection, object signing and signature verification, certificate management (including issuance and revocation), and other common PKI functions.

3.3 Opera Software

Opera Software ASA company develops two popular applications, the homonym browser, and the Opera Mail client (incorporated into the browser). The support for X.509v3 certificates, and the related cryptographic and PKI standards is a proprietary one.

3.4 Apple Safari

Safari browser still lacks a large part of security services available within the other browsers analysed during our investigation. In brief Safari relies either on the operating system cryptographic support, or on its partial proprietary implementation of security protocols and standards.

3.5 Google Chrome

The popular browser from Google is based on an open-source project called Chromium. Therefore, Chrome uses the system-specific implementation of SSL/TLS protocols, and the cryptographic libraries of each platform.

For Windows operating systems, Chrome uses Microsoft Secure Channel (SChannel) for SSL/TLS, and CryptoAPI. Secure Channel, also known as Schannel, is a security support provider (SSP) that contains a set of security protocols that provide identity authentication and secure, private communication through encryption. SChannel is used for Internet applications that require HTTPS-based communications.

For Unix/Linux, Chrome employs the security services implemented by Mozilla NSS.

For MacOS, Chrome uses Secure Transport for SSL/TLS and CSSM for crypto. The Common Security Services Manager (CSSM) is a shared library to which applications can link to obtain security services. It defines both the API and the service provider interface for add-in security service modules. CSSM includes a set of core services that are common to all categories of security services.

4. BROWSER-SPECIFIC PROBLEMS AND SOLUTIONS

One of the tests we ran in order to check the functionality of the application the PKI consisted in generating certification requests using combinations of most common browsers and operating systems used by our community of potential subscribers. The results observed during these tests for the most popular browsers installed on Windows XP Service Pack (SP) 2 machines are listed in **Table 1**.

Table 1. Browsers' behaviour with Windows XP Service Pack 2

Browser / Operation	Mozilla Firefox	Microsoft Internet Explorer	Google Chrome	Opera	Apple Safari
CSR generation	Works	Works	Key pair is server generated	Works	Key pair is server generated
Personal certificate installation	Works	Unable to validate the certificate	Unable to validate the certificate	Works	Unable to validate the certificate
Top-level certificate(s) installation	Works	Unable to validate the certificate	Unable to validate the certificate	Unable to install	Unable to validate the certificate

With Windows XP SP2 operating system, for three of the browsers listed above – Internet Explorer, Chrome and Safari – the operation of personal certificate installation is identical because all of them use the same underlying cryptographic library Microsoft CryptoAPI (2011). The user is guided throughout the installation by a wizard that asks the user into which repository the certificate should be stored. If the user chooses the default option, thus letting the wizard to automatically select the repository, the top-level certificates (and most importantly, the root certificate of ROCA) will go into the Intermediate Certification Authorities repository instead of the Trusted Root Certification Authorities one. In order to install them correctly, the user must explicitly select the installation of the ROCA's root certificate into the Trusted Root Certification Authorities repository.

Since Microsoft CryptoAPI does not support digital signatures employing SHA-2 digest algorithms by default, two approaches can be considered. The first option that seems easier to accommodate from the perspective of end users, consists in changing the digest algorithm used by the CA for the certificate signature to SHA-1, but this will weaken then the security of PKI due to collisions found in the algorithm according to Manuel (2008). The alternative is to update Microsoft CryptoAPI with the KB968730 (2011), or upgrading to Windows XP Service Pack 3.

Using Firefox 2.x, 3.x, 4.x and 8.x, the whole process works as expected: for CSR the only thing the user has to do is to complete the form fields; the personal certificate installation is done by means of a click on the hyper-link received via the notification e-mail. Then, the top-level certificates (root CA certificate included) can be easily installed by following the two hyper-links available on the website of ROCA (2011).

In Internet Explorer 7.x and 8.x, just before requesting a certificate, the user has to accept running an ActiveX code, by clicking *Allow* on the information bar at the top of the browser. The personal certificate installation is almost as straightforward as in Mozilla Firefox, the only difference being that a message informing the user about the site intention to execute an operation related to digital certificates must be allowed. The top-level certificates (root CA certificate included) can be installed following the two hyper-links available on ROCA website, using the Windows wizard.

With Google Chrome 13.x, 14.x and 15.x, the CSR generation works well at a first glance, but when the user will try to install the personal certificate, he will notice that the certificate along with the key pair are available for download on the CA website, being protected by the password she entered during the CSR generation process. The certificate will be then saved as a PCKS#12 file that can be installed by opening it from the Downloads section of the browser. The top-level certificates can be installed following the two hyper-links available on the website, using the Windows wizard.

In Opera 9.x, 10.x and 11.x during the CSR generation process, the user still has to adjust the key length, as the default value of 2048 bits from our application is replaced with a smaller value of 1536 bits. The certificate installation works as expected. The user can install her certificate by clicking the hyper-link received within the issuance notification e-mail, but problems arise during the installation of top-level certificates. The intermediate certification authority is correctly installed, but the root certificate can neither be downloaded nor installed.

For Safari 3.x, 4.x and 5.x, the CSR generation process has the same issues as for Chrome, the key pair being generated by certification authority. The personal certificate installation raises another problem, because the downloaded PCKS#12 file cannot be directly installed from the Downloads window, due to an error stating that no application is registered for this kind of files. In other words, Safari has no support so far for this public-key cryptographic standard. Nevertheless, the certificate can be installed if opening it from Windows Explorer. The top-level certificates can be installed following the two hyper-links available on the website, and using the Windows wizard.

As Microsoft CryptoAPI from Windows XP was replaced with Microsoft CryptoAPI NG (Next Generation) in Windows Vista and Windows 7, the behaviour related to cryptography has changed in some aspects for browsers based on CryptoAPI – Internet Explorer, Chrome and Safari. The change we noticed was related to the support added for NIST suite B algorithms (CNG 2011), especially for SHA-2 family (SHA-256, SHA-384 and SHA-512) of digest algorithms. Due to this update, the certificates signed with SHA-2 signature are no longer marked as corrupted or altered when importing them in browser, and they can be used successfully as personal or server certificates.

As shown in **Table 2**, Chrome, Safari, Opera and Firefox have almost the same behaviour as in Windows XP SP2, the only difference being the one stated in the above paragraph.

When using Internet Explorer 8.x and 9.x, things complicate a bit. To be able to make a request, one should configure Internet Explorer to accept and run ActiveX scripts marked as not safe, and then add the CA website used to request the certificate into the *Trusted Websites* list. Before the generation of the key pair, a window titled *Creating a new RSA exchange key* will pop up, and here the user should choose *High security level* along with a password used to protect the certificate's private key. The certificate can then be installed following the hyper-link received in the issuance notification e-mail from the CA, but when the web page opens, Internet Explorer will pop up an alert announcing the user that the web site is attempting to perform a digital certificate operation on her behalf. After explicitly allowing the operation to take place, the certificate will be installed into the repository. The top-level certificates can be installed following the two hyper-links available on ROCA website, using the Windows wizard.

Table 2. Browser behaviour for Microsoft Windows 7 operating system

Browser / **Operation**	**Mozilla Firefox**	**Microsoft Internet Explorer**	**Google Chrome**	**Opera**	**Apple Safari**
CSR generation	Works	Works	Key pair is server generated	Works	Key pair is server generated
Personal certificate installation	Works	Works	Works	Works	Works
Top-level certificate(s) installation	Works	Works	Works	Unable to install	Works

4.1 *Certificate Signing Request Generation*

Regarding the CSR generation using certification authority public interface, two different behaviours were noted, depending on the browser used. When using Firefox, Opera or Internet Explorer, it is respected the signification of the default setting regarding key pair generation, whose value is *Browser (Your Computer)*. In the case of the other two browsers, Chrome and Safari, even if on the interface the default setting is shown, at the completion of the certification subscription, the key pair will be generated by the certification authority, and its download by the user will be protected by the password introduced by the user in the requested PIN field when generating the request.

No matter what browser will be used, during the form’s completion for certification request it will be necessary to choose a PIN code, whose role is to protect the private key when this is generated by the certification authority. Only under this circumstance the PIN code will be requested every time when trying to download the key pair from the certification authority. When the key pair is generated by the browser, although it will also be necessary to complete the form field concerning the PIN code, the access to private key is actually restricted by the password introduced in the dialog *Creating a new RSA exchange key* when using Internet Explorer, or by the Master Password in Firefox or Opera. Because the PIN is sent in clear between the subscriber and the certification authority, the usage of a secure channel is mandatory in order to protect the communication between the two entities.

By analysing the packets exchanged between the CA and the user's browser after completing the generation of the certification request, the problem regarding the key pair generation in Chrome and Safari can be noticed promptly. In the HTTP POST request which follows after the user presses *Generate Request* button, no public key or PCKS#10 CSR can be traced, and the availability for download of the key pair from the certification authority indicates the fact that both keys were generated by the CA, and not by the browser, as intended. For the other three browsers tested, inside the HTTP POST request there is a field named either *newkey* – for Firefox and Opera – that contains the public key, or a field named *request* – as is the case with Internet Explorer – that contains the CSR in PCKS#10 format, including here also the public key.

4.2 Solutions for Top-level Certificates Installation

As it was shown earlier, user certificates can be installed following the hyper-link from the issuance notification e-mail, a click being the only action one should do. It cannot get any simpler. But for the top-level certificates, the procedure renders a different level of complexity depending of the browser used. To simplify the process even more, different combinations of scripts can be used.

For Internet Explorer, the Certificate Import Wizard can be totally avoided using a combination of VBScript and JavaScript scripts that will perform the installation of the certificate chain saved as PKCS#7 with the help of the Certificate Enrolment API implemented in Xenroll.dll or CertEnroll.dll for all Windows operating systems.

For Firefox and Opera, the installation can be simplified by sending the PKCS#7 file within a message with Content-Type set to *application/x-x509-ca-cert* that will force the browsers to start the installation dialogue immediately (as detailed in Mozilla Developer Network, 2010), as in the following example.

```
$filename = "roca.cacert";
$shortname = basename( $filename );
header("Pragma: ");
header("Cache-Control: ");
header("Content-type: application/x-x509-ca-cert");
header("Content-Disposition: attachment; filename =
\"".$shortname."\"");
header("Content-length:".(string)(filesize($filename)));
set_time_limit(0);
readfile($filename);
```

For the other two browsers – Chrome and Safari – a solution is still under research.

ACKNOWLEDGMENT

This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007-2013.

REFERENCES

Barth, A., Jackson, C., *The security architecture of the Chromium browser*, Google Chrome white paper, 2009, available on-line at http://seclab.stanford.edu/websec/chromium/chromium-security-architecture.pdf

Manuel, S., *Classification and generation of disturbance vectors for collision attacks against SHA-1*, International Association for Cryptologic Research, the Cryptology ePrint Archive, 2008, available on-line at http://eprint.iacr.org/2008/469.pdf

Marian, M., *A PKI Case Study for the Educational Environment*, Proceedings of 18th International Conference on Systems, Signals and Image Processing 2011, IWSSIP 2011, Sarajevo, Bosnia and Herzegovina, ISBN 978-9958-9966-1-0, IEEE catalogue no. CFP1155E-PRT, June 2011, pp. 409 – 413.

Microsoft Certificate Enrolment API, 2011, available on-line at http://msdn.microsoft.com/en-us/library/windows/desktop/bb427413(v=vs.85).aspx

Microsoft CryptoAPI and Cryptographic Service Providers, a Microsoft Technical Network white paper, 2011, available on-line at http://technet.microsoft.com/en-us/library/cc962093.aspx

Microsoft CryptoAPI Next Generation, Cryptography API: Next Generation (CNG), available on-line at http://msdn.microsoft.com/en-us/library/windows/desktop/bb204775(v=VS.85).aspx#suite_b_support

Microsoft Knowledge Base, *Windows Server 2003 and Windows XP clients cannot obtain certificates from a Windows Server 2008-based certification authority (CA) if the CA is configured to use SHA2 256 or higher encryption*, a Microsoft Support article, 2010, ID 968730, available on-line at http://support.microsoft.com/kb/968730

Mozilla Developer Network, *NSS Certificate Download Specification*, a Mozilla MDN white paper, 2010, available on-line at https://developer.mozilla.org/en/NSS_Certificate_Download_Specification

Romanian Certification Authority – ROCA, 2011, available on-line at http://ca.ucv.ro