Issuance of Proxy Certificate Residing In Web Portal By Browser Based Digital Certificate Through HyperText Transfer Protocol

Sea Chong Seak, Ng Kang Siong, Tan Fui Bee, Galoh Rashidah Haron

***Abstract*— Proxy certificate has been used in many web portal services, especially if the remote services require the services to act on user behalf to connect to other servers that require certificate based authentication. This paper describes the method of enabling direct creation of proxy certificate via web browser to web portal using user's private key that resides in client computer. This method can be used for proxy certificate issuance on any web based applications. For example, proxy certificate can use by web services portal for job delegation and single sign-on to multiple computing nodes.**

***Keywords*— PKI, Proxy Certificate, X.509, SSL/TLS, browser extension program**

I. INTRODUCTION

WEB portal consolidates information from multiple web servers before serving to the end user. The situation is clear-cut if the information is retrieved from publicly available web servers that do not require user authentication. In situation where the web portal consolidates information from web servers that require user authentication, the user need to deposit the authentication information (e.g. username-password pair, One-Time-Token) at the web portal first. The web portal then sends the authentication information to the web servers hosting the information to complete the user authentication process required by the web servers. As far as these web servers are concern, the information is release based on the user authentication information forwarded by the web portal.

In situation where each web server requires different user authentication methods, the web portal will act as a single sign-on portal to these web servers. The user needs to deposit all the user authentication information required by each web servers. Unfortunately depositing the user authentication information on the web portal creates security vulnerabilities.

In situation where the web server requires SSL certificate based authentication, the solution is not as straight forward. This is because user private key is required in the SSL certificate based authentication process. However, in order to maintain secrecy of the user private key, it cannot be deposited at the web portal. Solution to this problem is to use proxy certificate and its matching proxy private key at the web portal to perform SSL mutual authentication with the web servers. The user can generate proxy certificate using the user certificate and user's private key. In this way, the user's private key remains at the user, meanwhile the proxy private key is used to act on behalf of the user's private key. In this way, the vulnerabilities of depositing user authentication information are mitigated by using proxy certificate.

It is the purpose of this paper to describe a method to generate proxy certificate and its matching proxy private key at the web portal by using browser based user certificate through HyperText Transfer Protocol (HTTP) in SSL channel.

II. BACKGROUND

A. Proxy Certificate

The concept of proxy certificate is to allow a certificate and its matching proxy private key to take the role of the actual user certificate and private key [3]. Proxy certificate can successfully resume the role of a user certificate if and only if it can be determined that the proxy certificate originated from the user certificate.

Proxy certificate and its matching proxy private key are required in situation when the user's private key is not available to the system. Proxy private key and proxy certificate are created so that in the absence of the user's private key, the proxy private key can take the role of the user's private key to perform digital signature on the data provided as in the case of SSL certificate based authentication.

Verification of digital signature performed by the proxy certificate can be done using the proxy certificate which can later be traced back to the user certificate.

Typically the validity period of the proxy certificate is short in order to limit exposure time in the event that the proxy private key is compromised.

Certification Authority (CA) issues user certificate by signing user's public key and user's information using CA's private key. In a similar way, a user can issue proxy certificate by signing a uniquely generated public key together with the user's information using the user's private key as depicted in Figure 1.

Due to the fact that CA's private key signs user certificate and user's private key signs proxy certificate, chain of trust is established from CA Root Certificate to proxy certificate. In other words, we can trace the chain of trust from the proxy certificate to the CA. Therefore, we can determine that the proxy certificate is indeed originated by the user.

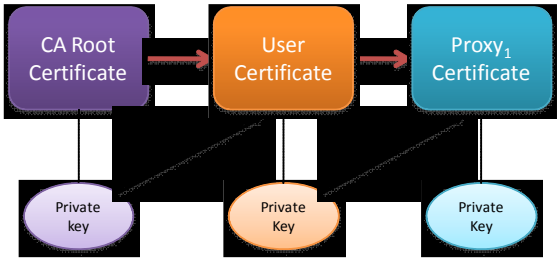

Figure 1: Chain of trust from CA to proxy certificate

In fact, subsequent level of proxy certificate can be generated by signing public key using proxy private key. Multiple levels of proxy certificates can be generated using similar method as depicted in Figure 2.

Proxy certificate finds its usefulness on single sign-on portal acting as front-end interface to subsequent servers that demand SSL mutual authentication.

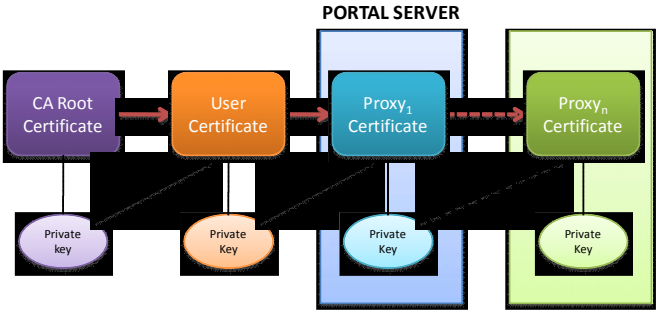

Figure 2: Chain of trust from CA to n number of the proxy certificates.

The proxy certificate normally has a rather short lifetime, typically 12 hours. In security term of a proxy certificate is compromised, the proxy private key must be treated carefully. Anyone who steals the proxy private key can perform any activity pretend to be authorized user.

Even though the proxy certificate validity is typically short, in hours, proper security measures are required to limit access to the proxy private key only to the web portal daemon.

B. SSL / TLS

Secure Socket Layer (SSL) and Transport Layer Security (TLS) [1] are commonly use protocol to establish encrypted channel between client and server. These protocols can also be used to ensure mutual authentication between client and server using digital certificates.

These protocols are supported by web browser and web server in the form of HTTPS [2], which essentially is HTTP over SSL/TLS secure channel.

HTTPS can also be established between web portal and web servers.

C. Web Server HTTPS Configuration

Default setup of web server such as Apache on HTTPS does not turn on the SSL certificate based mutual authentication. This means that client certificate at the browser is not required when connecting to web server via HTTPS by default. However, SSL certificate based mutual authentication can be enabled on most of the web servers so that only client certificates that are issued by trusted root certificate can be allowed to access the web server via HTTPS.

Most web server defines the location where the administrator can include root certificates that the administrator wants the web server to trust.

III. Problems

Limiting the exposure of proxy private key is necessary to maintain the secrecy of the proxy private for the duration of the proxy certificate validity. Constructing proxy certificate entirely at the user client browser by using the user's private key at the client browser is technically viable. However, transporting the proxy certificate together with the proxy private key to the web portal increases the exposure of the proxy private key. Therefore, generating the proxy public-private key pair at the web portal is a better security solution.

In order to avoid the transportation of proxy private key, there was an attempt to allow the user to generate proxy certificate at a credential storage server, e.g. MyProxy [8] via dedicated MyProxy client. A username-password pair is required by MyProxy server to authenticate the user. Whenever the web portal need a proxy certificate, the user deposit the username-password pair at web portal and later the username-password is sent to MyProxy server to instruct MyProxy server to generate proxy-proxy certificate based on the user proxy private key and certificate deposited at the MyProxy serve at the earlier stage. Unfortunately, this method requires the use of username-password pair which is another credential on top of the user certificate. The security risk for this method is similar to a standard single sign-on web portal that stores user's password.

IV. Solution

A preferable situation is to allow the user to use the same credential to authenticate to web portal and subsequently using the derived proxy certificate to authenticate the user to the web servers behind the web portal. The proxy certificate generation process should utilize existing HTTPS and browser so that no additional network port connection is required.

SSL mutual authentication is a standard feature provided by most of the browsers and web servers today. The user certificate and private key are accessible by the browser through the use of PKCS#11 [4] library for Mozilla based browser and CSP [5] library for Internet Explorer.

As for the mechanism to generate proxy certificate on the web portal, the entire process is broken up into two separate sub-processes, each running at web portal and browser computer respectively.

The tasks for the sub-process running on web portal are to generate proxy public-private key pair, construct unsigned proxy certificate and calculate proxy certificate digest. Meanwhile the task of the sub-process running on the user

computer is to sign the proxy certificate digest using user's private key.

Justification for such separation of task is to ensure that the proxy private key remains at the web portal and user's private key remains at the user side. No transportation of any private key is required and hence preserving the secrecy of private keys.

The next challenges are to identify the right implementation for above mentioned sub-processes within browser-web portal framework and to define the communication mechanism between these two sub-processes.

Common Gateway Interface (CGI) program is used to implement the sub-process running on web portal while browser extension program [6] is used to implement the sub-process running on web browser side.

Parameter passing between these two software components can be achieved by using two different methods. For sending parameter from web portal to browser, parameter is included within an embedded tag section in the HTML page. When the browser process the HTML page received from the web portal, the browser extension program is invoked to process all the parameter embedded within in embedded tag section. Meanwhile, for sending parameter from browser to web portal, the parameter is part of the information contain within the POST command sent by browser extension program to the CGI programming running at the web portal.

The solution can be described in the following steps:

(1) Generate proxy public-private key pair, construct unsigned proxy certificate and calculate proxy certificate digest at the web portal.
(2) Transport the proxy certificate digest to the browser.
(3) Sign the proxy certificate digest using the user's private key
(4) Transport the signed proxy certificate digest back to the web portal.
(5) Construct complete proxy certificate at the web portal.

Step (1) is achieved by CGI program running on web portal. CGI program is initiated by the user browser sending POST instruction to the web portal.

Step (2) is achieved by placing the proxy certificate digest in a parameter field within an embedded tag section in the HTML page generated by the CGI program. When the browser receives the HTML page from the web portal, the browser launches a specific browser extension program to interpret the embedded tag parameter. The proxy certificate digest in the embedded tag section is in URL encoded format.

Step (3) is achieved by using PKCS#11 software library for Mozilla browser while CSP software library for Internet Explorer to sign the certificate digest. PKCS#11 and CSP software library can be called directly by the browser extension program.

Step (4) is achieved by the browser extension program sending a POST command to a CGI programming running in the web portal. The signed proxy certificate digest is one of the parameters in the POST command string.

Step (5) is achieved by the CGI program running in the web portal.

V. Detail Proxy Certificate Generation Process

The user initiates the web browser and activates CSP for Microsoft Internet explorer or PKCS#11 for Mozilla Firefox to perform HTTPS SSL mutual authentication with web portal running Apache web server. The user certificate is verified by the web server. Only authorized user can login to the web portal.

In order to begin the proxy certificate generation process, user is required to enter the proxy certificate validation period (in hour) in a form on the web page. When the user clicks on the submit button on the HTML form, HTTPS POST command is sent to the web portal and this action activates the relevant CGI program to initiate proxy public-private key pair generation. Upon successful key pair generation, the CGI program stored the key pair in proper storage.

The next task of the CGI program after completing the key pair generation is to construct an unsigned X.509 format proxy certificate that complies with the requirement of IETF RFC 3820[3] for proxy certificate format. The user certificate has been transmitting to the web browser to the web portal via TLS/SSL digital certificate mutual authentication process at the earlier stage. The CGI program retrieves the user certificate from TLS/SSL session [7], and then extracts the necessary information from the user certificate to be used for the construction of the unsigned proxy certificate. The unsigned X.509 proxy certificate is constructed based on the above information and also inclusive the user enters validation period (in hour) and new generated pubic key.

The next step is for the CGI program to calculate certificate digest or hash value from the unsigned X.509 format proxy certificate. This certificate digest value is URL encoded and placed in a parameter field within an embedded tag section of the HTML page generated by the CGI program. The embedded tag used is as follows:

Embedded tag for Microsoft Internet Explorer

```
<OBJECT ID="IEProxyCert"
 CLASSID="CLSID:7D40EB7A-0A97-40C7-9669-
 CD70BA776E58">
<PARAM NAME="mDIGEST"
 VALUE="E015DB6A8ADA9985660B1E837AA8B078">
<PARAM NAME="mURL"
 VALUE="https://webportal.mimos.my/cgi-bin/
 cgisignedcert.cgi ">
<PARAM NAME="mTARGET" VALUE="_self">
</OBJECT>
```

Embedded tag for Mozilla Firefox

```
<embed type="application/pc-plugin" width=100
 height=50
 mDIGEST =" E015DB6A8ADA9985660B1E837AA8B078"
 mURL="https:// webportal.mimos.my /cgi-bin/
 cgisignedcert.cgi"
 mTARGET="_self">
```

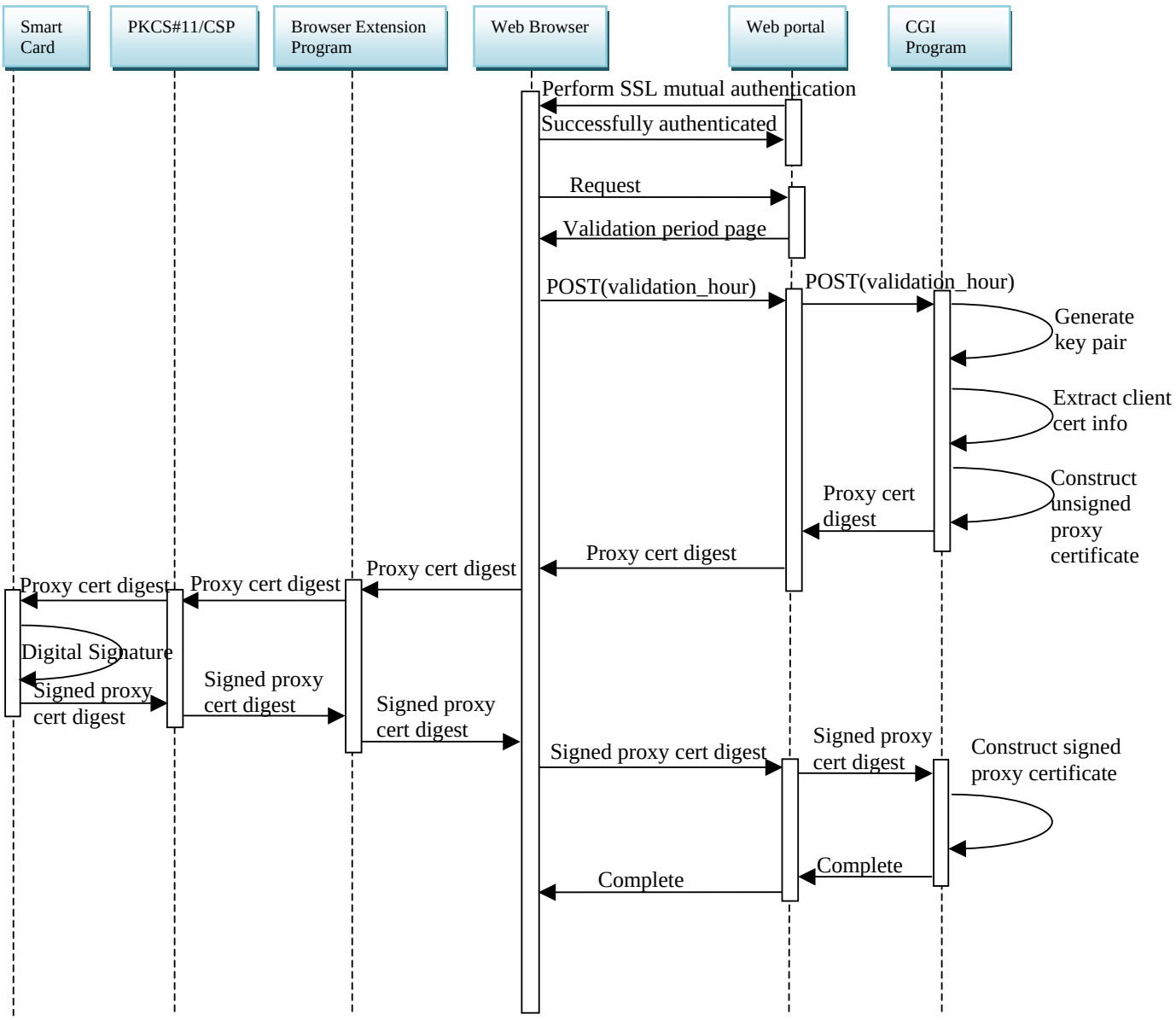

Figure 3: Sequence diagram for proxy certificate issuing process.

The mDIGEST parameter field contains the proxy certificate digest. The mURL parameter field is the target URL where the browser extension program will send the POST command to pass the signed certificate digest to the CGI program running on the web portal. The mTARGET parameter field is the target HTML frame name to receive the return HTML page from web portal after the POST command.

When the web browser receives the HTML page with the embedded tag containing the proxy certificate digest, the appropriate browser extension program that has been configured to associate with the browser is activated. The browser extension program is pre-installed on the user computer running with web browser and has been configured to associate itself with the plug-in application.

When the browser extension program received the proxy certificate digest, this digest is sent to PKCS#11 (Mozilla Firefox) or CSP (Microsoft Internet Explorer) library to be signed using the user private key. The signed proxy certificate digest is return to the browser extension program.

The final task of the web browser extension program is to initiate the web browser to send a POST command to deliver the signed proxy certificate digest to web portal via secure HyperText Transfer Protocol (HTTPS). This POST command and its payload of signed proxy certificate digest are received by the CGI program running at web portal. The CGI program constructs a complete proxy certificate based on the signed proxy certificate digest and the previously generated unsigned proxy certificate.

This concludes the entire proxy certificate issuance process between client web browser and web portal. The proxy certificate and its corresponding private key will be used by trusted web portal to initiate digital certificate based mutual authentication with other web server.

VI. Conclusion

We have described the detail process of issuing proxy certificate via web browser extension program with web portal and CGI program. We have also elaborated the needs and the advantages of using such method. One of the key advantages is to eliminate to use of other client server based application to issue proxy certificate.

This process is suitable for single sign-on web portal to login to certificate based authenticated servers on behalf of the users. Another potential usage of this process is to allow direct proxy certificate issuance on Grid Portal without using MyProxy server.

Further work is required to ensure a standardize message format for communication messages. A standardized format will allow wide spread usage of such process for general public.

References

[1] MDierks, T. and C. Allen, "The TLS Protocol version 1.0", IETF RFC 2246,January 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2246.txt

[2] E. Rescorla, “HTTP Over TLS”, IETF RFC2246, May 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2818.txt

[3] S.Tuecke, V. Welch, D.Engert, L. Pearlman, M.Thompson, Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate, June2004. [Online]. Available: http://www.ietf.org/rfc/rfc3820.txt

[4] RSA Laboratories, “Standards Initiatives: Public-Key Cryptography Standard (PKCS) #11: Cryptographic Token Interface standard”, version 2.20, 11 January 2007. [Online]. Available: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20a3.pdf

[5] Microsoft TechNet, “Microsoft CryptoAPI and Cryptographic Service Providers”, copyright 2008, Microsoft Corporation. [Online]. Available: http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dscj_mcs_xxgl.mspx?mfr=true

[6] Mozilla developer center, “Building an Extension”, Mozilla, [Online]. Available: http://developer.mozilla.org/en/docs/Building_an_Extension

[7] Ralf S. Engelschaal, “User Manual mod_ssl version 2.8”, mod_ssl.org, [Online]. Available:. http://www.modssl.org/docs/2.8

[8] Jason Novotny, Steven Tuecke, Von Welch, "An Online Credential Repository for the Grid: MyProxy," hpdc,pp.0104, 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 '01), 2001.