Public Key Infrastructure Operations Toolkit Based on Service-Oriented Architecture

Artem Harutyunyan
Yerevan Physics Institute,
Armenian e-Science Foundation
State Engineering University of Armenia
Yerevan, Armenia
hartem@mail.yerphi.am

Arsen Hayrapetyan
Yerevan Physics Institute,
Armenian e-Science Foundation
Yerevan, Armenia
ahairape@mail.yerphi.am

Gayane Kazhoyan
State Engineering University of Armenia
gayanek@gmail.com

Ruben Sefilyan
State Engineering University of Armenia
r.mitosh@gmail.com

ABSTRACT
The paper describes our work on the development of a modular toolkit based on Service-Oriented Architecture for the deployment of Public Key Infrastructure.

Keywords
Public Key Infrastructure (PKI), Certification Authority, Digital certificates, Service Oriented Architecture (SOA)

INTRODUCTION
The world-wide accepted standard for providing secure communication and authentication in networks is the Public Key Infrastructure (PKI) technology which consists of two essential ingredients: asymmetric (or public) key cryptographic technique and institute of Certification Authority (CA). CA is a body that issues to communicating parties digital certificates, which are trusted by all PKI participants.
In modern Grids and other scientific networks day to day CA work includes a number of operations: receiving certificate signing requests from End Entities (EEs) - users, hosts and services, issuing certificates for EEs, generating and publishing certificate revocation lists (CRLs), notifying EEs about their certificates expiration, etc. An important aspect of the CA work is providing its Registration Authorities (RAs) and users with appropriate software tools.
Currently available open-source software packages developed for running a CA are difficult to customize and extend.
The aim of this work is to create user-friendly, easy to extend and maintain open-source toolkit, based on service-oriented architecture, for effective certification process in PKI.
The toolkit suits the needs of classic CAs, members of International Grid Trust Federation [1], and will be suggested for their operations.

STATE OF THE ART
A typical cycle of a certification process in PKI involves users requesting certificates, RAs validating users' identities and approving the requests, CA issuing users' certificates. There are many open-source packages featuring these operations via graphical web interfaces, they can be categorized by their design patterns into following classes: i) applications which support many different types of possible CA configurations, ii) applications which support specific CA configurations. An example of an application of the first class is OpenCA [2] package, written in Perl programming language, which features possibilities to set up various CAs: from demo to high-level security on-line CAs. While practical for experienced CA managers it may be difficult for those with limited knowledge (especially of Perl) to setup a medium-level security CA (one usually required in Grids) quickly and easily, since this may require source code modification. It also restricts a developer who wants to extend the functionality of the toolkit to a particular programming language – Perl. PHPKI [3] is an example of an application of the second class. It supports CAs operating in trusted intranets, but does not allow maintaining medium (and high) security level required from CAs operating in Grids and large Public Key Infrastructures. This work intends to provide a toolkit which is easy to set up and use for creation of a medium-level Grid CA. In the meantime it provides extendable framework for developers for adding more sophisticated features.

REQUIREMENTS TO TOOKIT
The toolkit will satisfy the following requirements:
- Provide CA and RAs personnel with a web interface for carrying out their work
- Provide certificate requesters and holders with a web interface for managing their certificates and certificate signing requests
- Be customizable in a user-friendly way
- Be built from free and open source components
- Be modular: each operation (e.g. certificate issuance) has to be implemented as a separate module
- Be easily extensible: adding new modules, should not require modification of the existing code and should not to bind developer to a particular programming language
- Web interfaces should be accessible via popular Web browsers (Mozilla, Internet Explorer, etc.)

IMPLEMENTATION
To meet these requirements we have designed the toolkit on the base of the service-oriented architecture [4]. User interfaces construction and display functionality is consolidated in a so-called Framework (written in PHP), while modules implementing particular operations are connecting to the Framework and to each other using SOAP protocol [5, 6]. The use of SOAP allows implementing modules in many modern programming languages [7].

FRAMEWORK
Framework is used to communicate data between users of the toolkit and modules, which implement operations like certificate request or certificate issuance. Framework and modules, exchange data using SOAP protocol. This means that modules can be implemented in any language which supports SOAP: the data to be exchanged is formatted by the module as per SOAP specifications and sent to the Framework which converts them to PHP structures and processes them, and vice versa (Fig. 1).

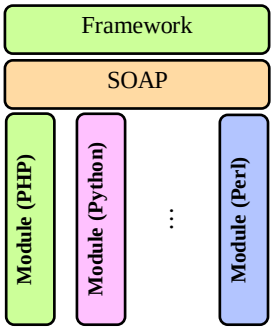

Fig. 1 Framework and modules

MODULES

The following modules have already been implemented:

- Internal module for accessing the functionality of the underlying OpenSSL cryptographic library
- Module for changing OpenSSL configuration files to issue certificates of different types (this allows implementing different grid certificate profiles for user, host and service certificates, as per [8])
- Module for initial setup of the CA (CA key-pair and certificate generation)
- Module for generating user certificate signing request
- Module for CA to sign certificate requests
- Module for converting the certificate to human-readable text format

All modules have to be registered with the Framework – this is done by adding a corresponding entry to the configuration file. Modules can be easily detached from the Framework removing corresponding entries in the configuration file.

SUMMARY

The toolkit is designed following principles of Service-Oriented Architecture, which enables developers to extend easily its functionality without restricting them to a particular programming language.

The current implementation of the toolkit provides functionality for users to request certificates, and for CA to perform basic operations like certificate signing, issuing CRLs, etc. The toolkit does not provide modules which will allow deployment the PKI with Registration Authorities (RAs). Implementation of RA modules requires introduction of the database to keep track of certificate requests and issued certificates. The work on the development of RA modules is currently underway.

It is also planned to develop modules for archival of the requests and issued certificates in the database, database back-ups and detailed logging of CA and RA operations.

ACKNOWLEDGEMENTS

We are very thankful to Ara A. Grigoryan for his help in paper preparation. The work of Artem Harutyunyan and Arsen Hayrapetyan was partially supported by Swiss Fonds ‘Kidagan’ and Calouste Gulbenkyan Foundation. This project has been supported by NATO grants NIG 983667 and CLG 983430.

REFERENCES

[1] International Grid Trust Federation – *http://www.igtf.net*

[2] M. Pala, "Generating Certificates for Grids with OpenCA 1.0.2+", *CAOPS Working Group, Open Grid Forum, Catania, Italy, March 2-6, 2009*

[3] S/MIME CA - *http://sourceforge.net/projects/phpki*

[4] Erl, T. “Service-Oriented Architecture: Concepts, Technology, and Design.” *Prentice Hall PTR, 2005*

[5] E. Newc, "Understanding Web services: XML, WSDL, SOAP, and UDDI", *Addison-Wesley, 2002, ISBN 0201750813, 9780201750812*

[6] SOAP Simple Object Access Protocol Specifications *www.w3.org/TR/soap/*

[7] SOAP implementations *http://www.soapware.org/directory/4/implementations*

[8]Grid Certificate Profile by European Policy Management Authority for Grid Authentication - *http://www.ogf.org/documents/GFD.125.pdf*