

Enhancing PHP Applications with Digital Signatures through SAPI Web Services

PHP via Web Services

The CoSign@ digital signature solution enables users and applications to digitally sign documents and data through a Web Services interface called SAPI Web Services. This Web-Services interface is based on OASIS Digital Signature Services (DSS) standard, which was announced in 2007 and is aimed at expanding application usage of digital signatures by setting definitions of how a digital signature application interacts with a centralized digital signature service (for more information, refer to www.oasis-open.org/committees/dss/ and www.oasis-open.org/committees/provision/).

One of the main benefits of SAPI Web Services is portability; there is a large variety of development environments and operating systems that can interact with it. In fact, any environment that supports SOAP (Simple Object Access Protocol) enhanced with TLS security, can benefit from using the digital signature capabilities of SAPI Web Services. Examples of such environments are Java-based applications that run on non-Windows platforms, pure .NET applications, and a variety of script languages such as PERL and PYTHON.

One of the leading script languages used extensively for Web application development is PHP. The PHP language has evolved greatly in recent years and it now enables Web developers to incorporate Web application login with minimal development efforts.

Since PHP includes SOAP capabilities and can also communicate securely with Web Services, it is simple to use CoSign from a PHP-based Web application through the SAPI Web Services interface. This document provides basic guidelines on how to use SAPI Web Services in a PHP environment, as well as a short coding sample.

Using CoSign with PHP

The following example describes how CoSign can be used with PHP. More detailed and technical information about using Web Services with PHP can be found in several existing documents, specifically Ayala, D. *et al.* Professional Open Source Web Services. Birmingham: Wrox Press Ltd., 2002. 305-328.

Please note that in order to successfully use CoSign, you will need the Web Services description file (.wsdl) of SAPI Web Services. Also, note that the following example uses PEAR, which is a high-level PHP module that allows interaction with a Web Service.

The example below performs the following actions:

- ▶▶ Creating a PHP class enabling the building of a buffer signature request.
- ▶▶ Calling the DssSign operation for performing the signature operation.
- ▶▶ Getting a response from the digital signature service.
- ▶▶ Displaying the response to the user.

```

<?php
try
{
    // INITIALIZATION SECTION
    require_once 'SOAP/Client.php';

    // initiating the SOAP client side based on the given WSDL file
    $wsdl_url = "SAPIWS-DSS.wsdl";
    $WSDL     = new SOAP_WSDL($wsdl_url);
    $client   = $WSDL->getProxy();

    // avoid validating the TLS Server certificate
    $client->setOpt('curl',CURLOPT_SSL_VERIFYPEER,0);

    // the data to be signed is base64 encoded
    $data = base64_encode("Hello World!");

    // PREPARING the SIGNATURE REQUEST

    // this namespace should be defined in several places
    $ns = "http://www.w3.org/2001/XMLSchema";

    // Authentication. In the case of Active Directory,
    // the domain name should be defined in the NameQualifier attribute
    $req->OptionalInputs->ClaimedIdentity->Name = new
        SOAP_Value('Name', $ns, "csnadm", array('NameQualifier'=>'arx'));
    $req->OptionalInputs->ClaimedIdentity->
        SupportingInfo-LogonPassword = new
        SOAP_Value('LogonPassword', $ns, "12345678", array('xmlns' =>
            "http://arx.com/SAPIWS/DSS/1.0"));

    // Signature Type
    $req->OptionalInputs->SignatureType = "urn:ietf:rfc:3369";

    // Data
    $req->InputDocuments->Document->Base64Data = new
        SOAP_Value('Base64Data', $ns, $data,
            array('MimeType'=>'application/octet-string'));

    $DssReq = new SOAP_Value('SignRequest', '', $req, array('xmlns'=>
        "urn:oasis:names:tc:dss:1.0:core:schema"));

    // PERFORMING SIGNATURE OPERATION
    $output   = $client->DssSign($DssReq);

    // RESULTS of the SIGNATURE OPERATION

    echo "result is: ";
    echo $output->Result->ResultMajor;
    echo "\n";
    echo $output->Result->ResultMinor;
    echo "\n";
    echo $output->Result->ResultMessage;
    echo "\n\n";

    $sig = $output->OptionalOutputs->DocumentWithSignature->
        Document>Base64Data;

    echo $sig;
    echo "\n";
}
catch (Exception $e)
{
    echo "Error!<br />";
    echo $e -> getMessage ();
}
}

```