Introduction

Library for digital signatures

Anonymous · 09/09/2026

Library for digital signatures

A comprehensive PHP library for creating and validating digital signatures, with special focus on XML digital signatures (XML-DSIG).

Features

- **Digital Signatures**: Sign and validate any data with RSA key pairs.
- **XML Signatures**: Full support for XML Digital Signatures (XML-DSIG).
- **Signature Verification**: Validate signatures against public keys.
- **Reference Support**: Sign specific sections of XML documents using ID references.
- **Integration**: Works seamlessly with Derafu Certificate and Derafu XML libraries.

Installation

```
composer require derafu/signature
```

Basic Usage

Signing Data

```
use Derafu\Signature\Service\SignatureGenerator;
use Derafu\Signature\Service\SignatureService;
use Derafu\Signature\Service\SignatureValidator;
use Derafu\Xml\Service\XmlDecoder;
use Derafu\Xml\Service\XmlEncoder;
use Derafu\Xml\Service\XmlService;
use Derafu\Xml\Service\XmlValidator;

// Set up the signature service.
$xmlEncoder = new XmlEncoder();
```

```
$xmlDecoder = new XmlDecoder();
$xmlValidator = new XmlValidator();
$xmlService = new XmlService($xmlEncoder, $xmlDecoder, $xmlValidator);

$signatureGenerator = new SignatureGenerator($xmlService);
$signatureValidator = new SignatureValidator($signatureGenerator, $xmlService);
$signatureService = new SignatureService($signatureGenerator, $signatureValidator);

// Sign simple data.
$privateKey = '...';
$data = 'Hello, world!';
$signature = $signatureService->sign($data, $privateKey);

// Validate the signature.
$isValid = $signatureService->validate($data, $signature, $publicKey);
```

Signing XML Documents

```
use Derafu\Certificate\Service\CertificateLoader;

// Load a certificate.
$certificateLoader = new CertificateLoader();
$certificate = $certificateLoader->loadFromFile(
    '/path/to/certificate.p12',
    'password'
);

// Load XML to sign.
$xml = file_get_contents('document.xml');

// Sign the entire XML document.
$signedXml = $signatureService->signXml($xml, $certificate);

// Sign a specific element in the XML document (identified by ID).
$signedXml = $signatureService->signXml($xml, $certificate, 'elementId');

// Save the signed XML.
file_put_contents('signed_document.xml', $signedXml);
```

Validating XML Signatures

```
use Derafu\Signature\Exception\SignatureException;

// Load signed XML.
$signedXml = file_get_contents('signed_document.xml');

try {
    // Validate the XML signature.
    $signatureService->validateXml($signedXml);
```

```
    echo "Signature is valid!";
} catch (SignatureException $e) {
    echo "Signature validation failed: " . $e->getMessage();
}
```

Advanced Usage

Detailed XML Signature Validation

For more detailed control over the validation process:

```
// Create a signature node from the signed XML.
$signatureNode = $signatureService->createSignatureNode($signatureXml);

// Validate the digest value (integrity of the signed content).
$signatureService->validateXmlDigestValue($xmlDocument, $signatureNode);

// Validate the signature value (authenticity of the signer).
$signatureService->validateXmlSignatureValue($signatureNode);
```

Calculating Digest Values

```
use Derafu\Xml\XmlDocument;

// Load XML document.
$xmlDoc = new XmlDocument();
$xmlDoc->loadXml($xml);

// Calculate digest value for the entire document.
$digestValue = $signatureService->generateXmlDigestValue($xmlDoc);

// Calculate digest value for a specific element.
$digestValue = $signatureService->generateXmlDigestValue($xmlDoc, 'elementId');
```

XML-DSIG Implementation Details

The library implements XML Digital Signatures according to the W3C XML Signature Syntax and Processing specification:

1. The `Signature` element is created with the following components:
    - `SignedInfo`: Contains information about what was signed.
    - `SignatureValue`: Contains the actual signature value.
    - `KeyInfo`: Contains information about the key used to validate the signature.

2. Canonicalization is performed using the C14N algorithm (http://www.w3.org/TR/2001/REC-xml-c14n-20010315).

3. Signatures are created using RSA-SHA1 (http://www.w3.org/2000/09/xmldsig#rsa-sha1).

4. Digests are created using SHA1 (http://www.w3.org/2000/09/xmldsig#sha1).

XML-DSIG Structure

When signing an XML document, the resulting signature will have the following structure:

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="#elementId">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>...</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <RSAKeyValue>
        <Modulus>...</Modulus>
        <Exponent>...</Exponent>
      </RSAKeyValue>
    </KeyValue>
    <X509Data>
      <X509Certificate>...</X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
```

Integration with Other Derafu Libraries

This library is designed to work seamlessly with other Derafu libraries:

- **Derafu Certificate**: For handling digital certificates and key pairs.
- **Derafu XML**: For handling XML documents and operations.

Last updated on 09/09/2026