Validation

Validating XAdES signatures: the result of each signature and the issues of its properties

Anonymous · 09/11/2026

Validation

`XadesValidator::validate()` validates the signatures of a document and checks the XAdES properties of each one. The signatures are validated by `derafu/signature` (`SignatureValidatorInterface`), that the validator receives.

```
$report = $validator->validate($signed);          // A string or an
XmlDocumentInterface.

$report->isValid();                                // Valid signatures and conformant
properties.
$report->areSignaturesValid();                     // Only the signatures.
$report->areConformant();                          // Only the properties.

foreach ($report->getSignatures() as $result) {    // One result for each signature.
    echo $result->isValid() ? 'valid' : $result->getError()?->getMessage();
}

foreach ($report->getIssuesOf(0) as $issue) {      // The issues of the first
signature.
    echo $issue->getMessage(), "\n";
}
$report->getIssues();                              // The issues of all of them.
```

An issue is an `XadesException` that is returned, not thrown. A signature that is not valid is a result, as in `derafu/signature`; the properties that do not follow XAdES are issues. The two are independent: a document that was changed has an invalid signature, and its properties can be conformant.

What Is Checked

For each signature:

- It has `QualifyingProperties` in one of its `Object`.
- The signature has an `Id`, and the `Target` of the properties is `#` and that `Id`.
- The properties have `SignedProperties`, with an `Id`, and the signature signs them: it has a `Reference` of the type `http://uri.etsi.org/01903#SignedProperties` to `#` and that `Id`.
- The `SignedSignatureProperties` have a `SigningTime` that is an `xsd:dateTime`.

- The signed properties have the certificate of the signer (`SigningCertificate` or `SigningCertificateV2`, each checked as its version says), and the first `Cert` is the certificate of the signature:
    - its `CertDigest` is the digest of the certificate (its algorithm has to be one of those that `derafu/signature` supports);
    - in `1.3.2`, its `IssuerSerial` has the serial number and the issuer of the certificate (the issuer is compared as a distinguished name, so the order and the format of its attributes do not matter);
    - in `1.3.2-201601`, if it has `IssuerSerialV2`, it is the issuer and the serial number of the certificate in DER.
- The `ObjectReference` of each `DataObjectFormat` and `CommitmentTypeIndication` is `#` and the `Id` of a `Reference` of the signature.
- The `QualifyingProperties` are valid for the official schema (XSD) of ETSI of at least one version, that the package has in `resources/schemas/`. Each error of the schema is an issue, that names the version of the certificate of the signer (the default one if the properties do not have it). The check is on by default; the third argument of the constructor (`checkSchema: false`) turns it off.

```
$validator = new XadesValidator($signatureValidator, new VersionRegistry(),
checkSchema: false);
```

What Is Not Checked

The hash of the signature policy is not compared with its document (the document is not in the signature), the roles of the signer are not checked, and the levels above B-B (time-stamps and the data of revocation) are not looked at. The trust in the certificate of the signer (that an authority issued it and that it is not revoked) is not in the signature either: it is decided with the certificate (`derafu/certificate`).