

# XML Electronic Signatures

The XML Electronic Signature is designed to be a very flexible signature model due to the inherently unconstrained nature of XML data and the variety of applications for XML. XMLDSIG, the initial instantiation of the XML electronic signature, relies heavily upon digital signatures as indicated by its name and the opening line of the [RFC 3275](#) abstract: “This document specifies XML (Extensible Markup Language) digital signature processing rules and syntax.” It was created to serve multiple masters – as a way to secure electronic “documents” as well as XML web service transactions and protocols.

The flexibility and extensibility of these rules is demonstrated by extensions to the XMLDSIG specification embodied in the [XAdES](#)-? specifications released by W3C and ETSI. There is a good [interview](#) with Donald Eastlake at the IBM Developer Works website discussing the issues of maintaining flexibility and the competing camps involved in the early standards work.

We will discuss the XMLDSIG extensions and XML digital signature's application to creating legally binding electronic signatures later in this document.

This extensibility creates the potential for extremely complex implementations, but they are based around a simple and elegant concept at the core of the XMLDSIG. The concept is that an XML signature will comply to XML specifications, but it can be used to sign arbitrary data in any format, inside an XML document or detached, as long as it can be referenced by a URI. This document is intended to describe XML signatures at a high level and to illustrate some of the issues that arise when trying to create electronic signatures that meet the requirements of ESIGN and the European Directive on electronic signatures.

Figure 1 shows an *enveloping* XML signature, it is useful for illustrating the concept. An XML `<Signature>` element is basically comprised of `<SignedInfo>` and a `<SignatureValue>` element. It optionally contains an `<Object>` as shown in Figure 1.

This is considered an enveloping signature because the data that is signed is contained inside the `<Signature>` element. If the `<Reference>` URI attribute points to data outside of the XML Document, then the signature is called a *detached* signature, if it points to an XML document element that contains the `<Signature>` then it is considered an *enveloped* signature.

`<SignatureValue>` is the digital signature value of the `<SignedInfo>` element. `<SignedInfo>` is the only XML that must be digitally signed (signed with a digital signature). The `<SignedInfo>` element contains the `<Reference>` element, this contains a pointer to the data being signed and the message digest of the signed data. The referenced data can be anything, a binary

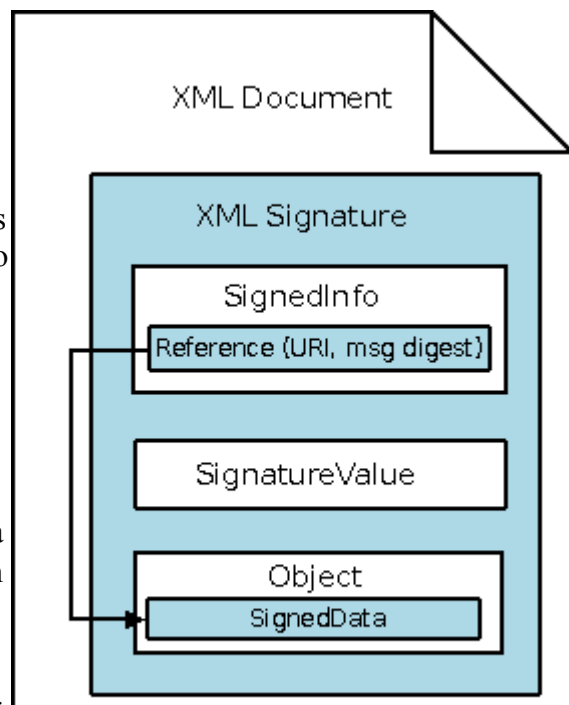


Figure 1. Enveloping XMLDSIG

software file, a movie or a pdf document file.

To verify the XMLDSIG the receiving application needs to verify that the `<SignatureValue>` element is the digital signature of `<SignedInfo>`. If the verification is successful (`<SignedInfo>` is valid) then it reads the data pointed to by the `<Reference>` URI, in our diagram this is the `<SignedData>` element inside of `<Object>`. The application calculates the message digest of `<SignedData>` and compares it to the message digest value element in `<Reference>`. If the message digests match the relying party can be confident that the signed data has not been modified after it was signed by the agent with the digital signatures private key.

After verifying the authenticity of the data the receiving party needs to validate the digital signature itself. If the digital signature relies on a digital certificate, is the certificate valid? Was it issued by a trusted authority? Is the signature's owner authorized to sign this transaction? Similar considerations can be applied to other optional characteristics of the `<SignedInfo>` and `<SignatureValue>` elements.

Analysis of the XML Electronic Signature provides an excellent illustration of the shortcomings of the digital signature alone to adequately address all of the issues of complying with E-SIGN. The XMLDSIG document does not have to follow the E-SIGN compliant process described in the implementation section. In fact, the XMLDSIG document doesn't even have to be signed by a digital signature at all, a compliant XMLDSIG document can be generated relying on the secret-key based HMAC algorithm, thus confounding the idea of associating a single individual or agent with the document altogether.

Likewise there is no specified provision within the XMLDSIG standard for affixing signature date and time or other such requirements that may be required when executing a valid wet-ink signature.

## **XAdES**

The European Telecommunications Standards Institute (ETSI) created the XML Advanced Electronic Signature (XAdES) standard, ETSI TS 101 903, that extends XMLDSIG to comply with the European Directive for electronic signatures. It strengthens the non-repudiation characteristics and adds signature policies, trusted time stamps and long term validation capability to the validation of the digital signature. This has formed the basis for a note by the W3C with minor changes.

XAdES adds “Qualifying Properties” to XMLDSIG. To generate the basic XAdES an `<Object>` element is used to contain (or reference) a `<SignedProperties>` element which must contain a `<SignedSignatureProperties>` child which must contain:

- A reference to the signer's certificate used for the signature
- A reference to the signature policy in effect at the time
- The time that signer claims to have produced the signature

Optional `<SignedSignatureProperties>`:

- The location that the signature was produced
- The role of the signer

Additionally there may be an optional `<SignedDataObjectProperties>` element which may contain the following children:

- Data object format
- Commitment type indication (depending on signature policy)
- All data objects time stamp (to demonstrate that the signature was created after this time)
- Individual data objects time stamps

There is also an optional unsigned element, the `<CounterSignature>` which is an XAdES or XMLDSIG signature that can be used to sign the `<Signature>`.

All of these Qualifying Properties are to increase the reliability of the signature, to further clarify its meaning and protect against later repudiation. And there is more:

XAdES-T adds a trusted timestamp over the entire `<Signature>` element to establish that it existed before a certain time. This is supposed to assist with non-repudiation based upon compromise of a signing key.

XAdES-C (for Complete) is XAdES-T with the addition of references to the validation data used to validate the signature. This is typically a reference to a chain of certificates in the validation path along with CRL or OCSP tokens from each issuer showing that the certificates have not been revoked.

XAdES-X (for Extended) is an XAdES-C with a time stamp over all of the validation data to demonstrate that it was validated before a certain time. This is starting to get to be a big structure, but there is more!

XAdES-XL is an XAdES-X designed for long term validation. Over time the references included in XAdES-C may become unusable. So XAdES-XL requires that all of these be dereferenced and the actual validation data included.

Finally there is XAdES-A, for Archival. This uses a trusted timestamp over the entire XAdES-XL and stipulates that the signature algorithm used should be more advanced/secure than the original time-stamping and signing algorithms used. XAdES-A is to be applied to existing archival copies as the strength of the signing algorithms becomes suspect to insure that the documents can't be changed by compromising the original signatures sometime in the (relatively) distant future.

## **MISMO Smart Doc™**

The US based Mortgage Bankers Association chartered the MISMO group to take on the task of creating ESIGN and UETA compliant signed electronic documents for the mortgage industry. MISMO also chose to use XMLDSIG as the basis for their electronic signature. Smart Doc is an industry specific XML based document format, the specification is available at [MISMO.org](http://MISMO.org).

MISMO faces a unique set of challenges because of the number of corporate and government entities that have to deal with the electronic records created. These include lenders at every level, “government sponsored entities” (Ginnie Mae, Freddie Mac), county recorders, insurance companies, attorneys, title agents and home buyers.

The basic idea of Smart Doc is that the root XML document contains 5 major sections: Header, Data, View, Audit Trail and Signature. This paper will not attempt to deal with the intricacy of the MISMO specification, suffice it to say that the flexibility of XML and XMLDSIG allows the Smart Doc to be tailored to meet these very unique requirements.

Smart Doc signatures require time stamping, role, and many other properties for signers. Smart Doc

uses the basic XMLDSIG construct to create a “tamper evident seal.” Smart Doc also allows signatures that exist in XHTML within the View section. Signature types allowed in Smart Doc are Text (which will represent basic typing or click-through type signatures), Object (can represent recorded handwritten signature images, or any other type that will use an independent applet to record), Digital Signature or Other.

Smart Doc includes the Audit Trail section within the document to record the creation, modification and signing process. A record of this nature is crucial for meeting the E-SIGN criteria for legally binding electronic documents, whether this information is embedded in the document, the signature or in a separate record is a design decision that will need to be made by the implementers of the electronic signing system.

One category of Smart Doc has seen some adoption as a negotiable instrument in the secondary mortgage securities market. The current Smart Doc specification is at revision 1.1 but work is ongoing on version 3 (not a typo, don't ask).

## **Back to Business**

Can businesses make use of these standards?

XAdES assumes the existence of trusted authorities and depends upon the ubiquity and reliability of a PKI infrastructure, these are probably risky assumptions for many organizations. Even if the infrastructure exists in the selected business environment, XAdES still leaves it to the implementer to insure that the E-SIGN compliant signatures are generated. This will likely require further extensions to the XAdES schema or to the document schema to incorporate more evidentiary data on the process.

MISMO Smart Doc is an industry specific XML document standard, with its own set of challenges and it is not widely applicable. It does represent the efforts of a large group of financial institutions, vendors, service providers government representatives and government agents and can be used as a learning tool.

An item that hasn't been explicitly addressed by the XML signature and security community at large is the reliance on electronic credentials. While basing signatures on electronic credentials makes sense today for many business to business or business to government transactions, a universal electronic credentialing system for individuals does not yet exist.

Businesses address this in multiple ways. For relatively low value transactions, such as most Internet credit card transactions, the possession of a credit card and knowledge of the owner's billing address may suffice for electronic credentials. For higher value transactions, many businesses incorporate witnessed signatures into the electronic signing process. This means that the business provides a method for the consumer to create the electronic signature (possibly a digital pen) and then the consumer executes the signature in front of a representative of the business. The representative is then responsible for insuring the identity and authority of the individual executing the transaction and he or the signing system is the holder of the electronic credentials.

For many commercial transactions long term validation may be a concern. The lengths that ETSI has gone to insure the long term validation of digital certificates is encoded in the XAdES schemas. The inclusion of validation data within the document is an idea that every signature system can use, although most will have much more modest requirements than the full XAdES-A.

Likewise, maintaining an audit trail, as is done by Smart Doc, is a key requirement. The implementor

of electronic signatures is not only responsible for insuring that an appropriate process is followed, but also needs to insure that this can be demonstrated in a court of law!

Can XMLDSIG or XAdES meet the requirements for general document signatures?

The answer is probably not in their basic form. They will require extension or integration into a more comprehensive document schema such as Smart Doc to be used for legally enforceable electronic documents.

The beauty of the XMLDSIG standard is that many document signing scenarios can be satisfied by extending the XMLDSIG and XML document schema. The key to creating a legal transaction is not in the data structures used but in the documented process that is followed and can be audited. XMLDSIG simply saves the implementer some of the work of inventing the data structures!