SWAD-Europe: Deliverable 11.1: Framework for Security and Trust Standards

Project name:
Semantic Web Advanced Development for Europe (SWAD-Europe)
Project Number:
IST-2001-34732
Workpackage name:
SWAD-Europe:
Workpackage description:
http://www.w3.org/2001/sw/Europe/plan/workpackages/live/esw-wp-11.html
Deliverable title:
Framework for Security and Trust Standards
URI:

Authors:
Graham Klyne, Ninebynine.org.
Brian Matthews, CCLRC.

Abstract:
This paper examines some current standards for data exchange formats and protocols related to security, authorization policy and trust management.

The goal of this paper is to explore the interaction between various security- and trust-related mechanisms, and to identify enough of the purpose and content of these existing standards to indicate how they might be integrated in a Semantic Web trust and policy management framework.

Status:
First draft: 2002-12-18
Deliverable release: 2004-06-15
Comments on this document should be sent to the public SWAD-Europe mailing list, public-esw@w3.org,

---

1. Introduction -

The purpose of this paper is to explore a number of standard data exchange formats and protocols that convey information expected to be significant to a semantic web based trust and policy management framework, and to sketch a framework within which they might be integrated.

Some of the material here is taken from an earlier memo prepared for the iTrust working group, which surveys a number of protocols and data interchange formats that have significance for trust management.

To be effective, a framework for establishing trust between parties must be based on agreed protocols to exchange information on which trust decisions may be based. This in turn calls for broadly accepted standards. Obtaining the consensus needed for a technical specification to become a standard is very much easier if existing relevant standards work is used to the maximum extent possible.

The essence of Internet standards is to achieve interoperability and scalability for data communication services.

For interoperability, it is particularly important that communicating parties do not have to activate some kind of bilateral out-of-band agreement in order to start communicating with each other. This issue is particularly significant for establishing trust: if any form of bilateral arrangement already exists then one of the biggest difficulties of establishing trust has already been solved. Both interoperability and scalability are served by keeping designs as simple as possible consistent with performing the required functions.

Scalability is further served by minimizing the dependence on a single authority, and restricting any such dependency to purely technical matters. Experience with PKI deployment (or non-deployment) shows that trying to force all users into a single operational model of trust management just does not work at Internet scale. Whatever the participants may wish for, there will be variations in national, corporate and personal policy that a scalable system must accommodate.

Successful Internet standards are generally made up from a number of simple interlocking pieces. This approach seems to permit common implementations of shared technical requirements, while allowing for local policy preferences. But we should not forget that interoperability requires that all of these interlocking pieces can work with each other. One of the strengths of semantic web technology is that it provides a framework within which diverse information from a variety of sources can be brought together for some greater purpose.

1.1 Structure of this paper

Section 2 develops a framework for considering how protocols work together to achieve the goals of trust. It starts by analyzing some of the terminology associated with trust and security, and shows how the ideas thus described work together in the context of a simple data transfer. It then goes on to suggest that more complex transactions can be composed from simple transfers.

Section 3 discusses the role of RDF in a framework that uses and conveys trust information.

The following sections examine a range of security related data format and protocol standards, with a brief explanation of their role and showing how they fit in the trust handling framework sketched in section 2.

---

2. Interactions between autonomous communicating parties -

┤ TOC&nbsp;

A number of interrelated influences bear on interactions between autonomous communicating parties: trust, provenance, security, authorization, authentication, confidentiality, dependability, privacy, authorization, access control, exposure control, to name a few key ideas. Here is a characterization of some of these terms:

Security
: *a goal, bad things don't happen*

Dependability
: *a goal, good things do happen*

Privacy
: *a goal, personal information is not disclosed or abused*

Policy
: *rules for behaviour*

Provenance
: *information (metadata) about the source of some piece of data*

Trust
: *belief in (expectation of) the behaviour of a party for some given purpose*

Authorization
: *a decision about use of some mechanism or resource for a particular purpose*

Authentication
: *a mechanism, a receiver confirms the identity of a data sender*

Encryption
: *a mechanism, a sender restricts who can receive data*

Access control
: *a mechanism, controls access to a resource*

Exposure control
: *a mechanism, controls delivery of data to a recipient*

Negotiation
: *a mechanism, determines what data is transferred*

There are some related concepts that have a secondary role, in that they are ultimately subsumed by one of the concepts noted above:

- knowledge
- belief/expectation
- identity
- intent
- competence

2.1 A trust framework for simple data transfer

How do these leading concepts relate to each other in systems that deal with trust and security? In the context of communicating computer systems, everything reduces to simple transfers of information. Provision (or disclosure) of information is the key to all other things. A striking example of this is a private key: ultimate control over many resources is determined by knowledge of one.

A simple transfer of data is between two parties, a sender and a receiver, and includes the following key steps:

- preparation of data
- transfer a copy of the prepared data
- use the copy of data received

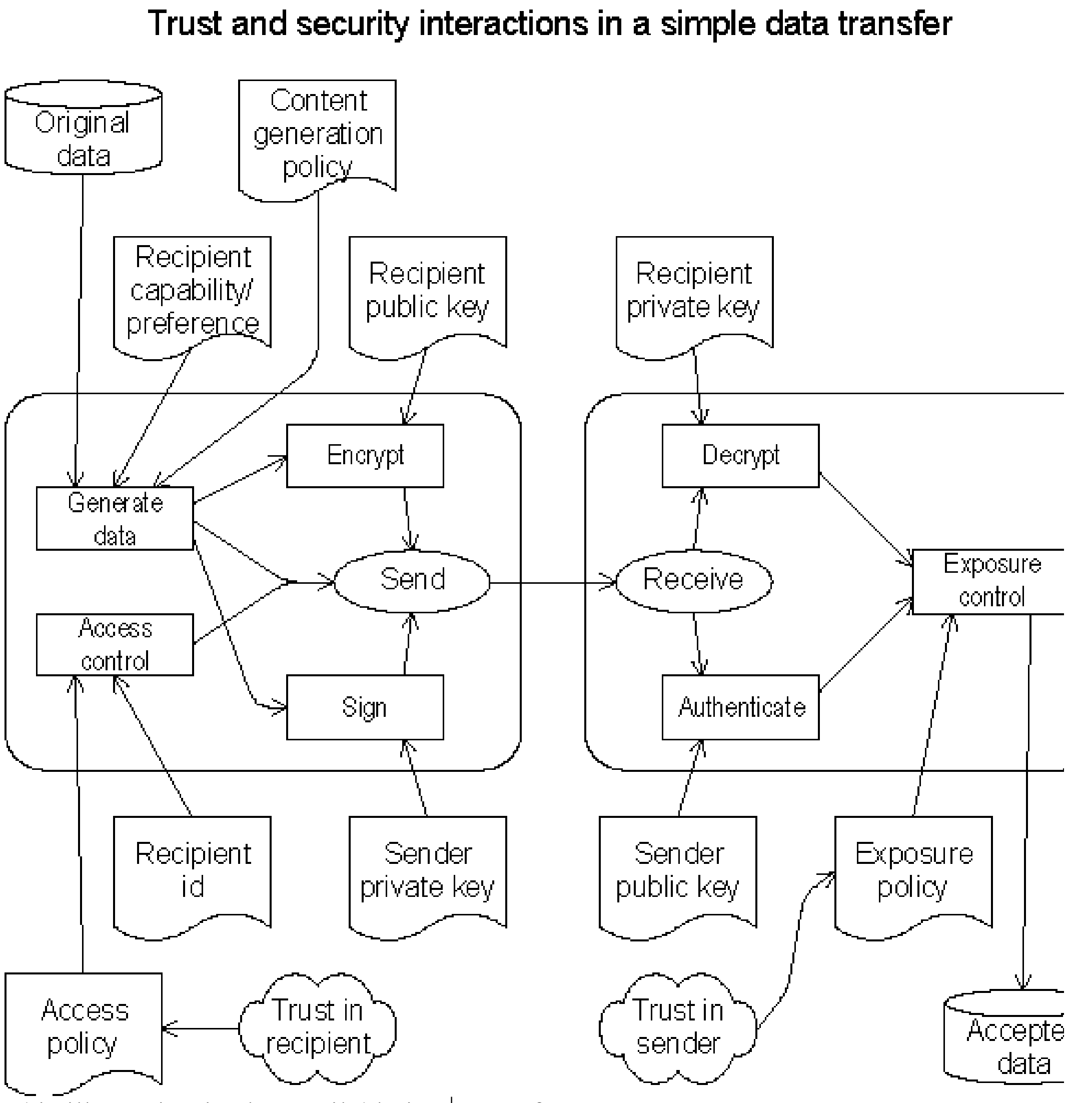
More complex transactions can be composed from such simple data transfers (see example below).

The sender may wish to be constrain the use of data provided (e.g. for reasons of confidentiality, privacy), or may wish to be confident that the data is usable and used as intended (e.g. serving a notice):

- access control can be used to restrict the sending of data, with authorization decisions based on authentication, provenance and policy information,
- encryption can be used to restrict who can receive the raw data, using policy information and knowledge of the intended recipient to guide the encryption process,
- negotiation can be used to obtain assurance that any data sent will be properly understood and usable, using provenance and policy information and knowledge of the intended recipient to select the data sent,
- the sender must trust that the recipient will honour any restrictions on the use of any data transferred, such restrictions being indicated through provenance and policy information.

The receiver may wish the origin and kind of data received to be limited (e.g. requiring authenticity, acceptable content and a usable format):

- authentication can be used to confirm the provenance of data,
- exposure control can be used to restrict data received, with authorization decisions based on content inspection, provenance and policy information,
- the receiver may need to trust that the data received was appropriately originated consistent with any associated provenance information.



This illustration is also available in PDF format.

In conclusion, I find that authentication, provenance, trust and policy are used as input to authorization decisions, and also to guide the use of encryption and negotiation mechanisms. Authorization decisions control the enforcement of access control and exposure control.

I also note that, even when all the indicated mechanisms have been employed, satisfaction of the intended transfer conditions usually still depends on the receiver's treatment of the received data. This is an unavoidable element of required trust that is difficult to completely eliminate.

2.2 Example of an extended transaction

In this example, we consider requested access to a resource with privacy and content negotiation.

In the following, R is the receiver of transferred data and S is the sender:

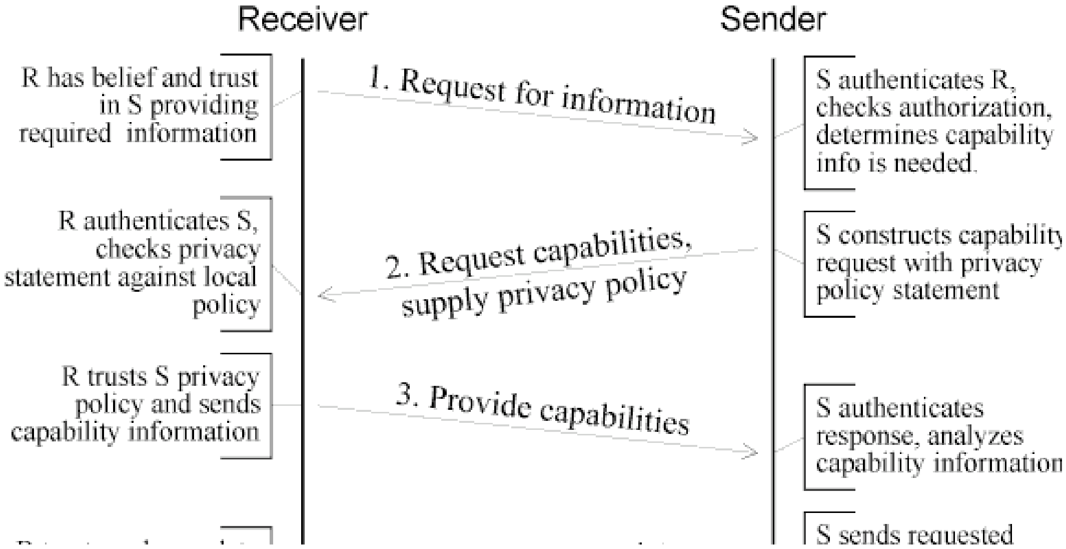
1. R requests S to provide some information.
2. S requests information about capabilities of R, providing information about their privacy policy with that request.
3. R provides description of capabilities.
4. S selects and sends data.

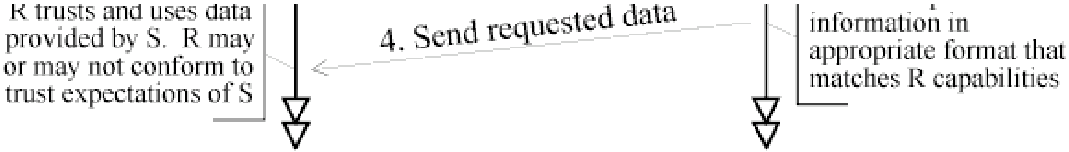
Each of these steps corresponds to a simple data transfer, of the form described above:

1. R assumes or has prior knowledge that S can understand the form of the

request, and can provide the required information. R trusts that S will not make inappropriate use of the fact that R has requested the resource (or does not care). The request might be encrypted to prevent disclosure to parties other than R or S. S makes few assumptions about the content of the request, and screens it carefully to ensure that it is a recognized, valid request. R may choose to authenticate the request data, and S may choose to check its authenticity.

2. If authenticity of the request is confirmed in step 1, S may check that R is authorized to receive the requested information. S also determines that one of several different forms of data may be provided, depending on the capabilities and/or preferences of R. S constructs a request for capability and preference information, including a description of the policy that S will use to govern use of that information. S probably has no concerns to constrain use of this policy information, so can assume no inappropriate use will be made. Based on the form of the request and other information from step 1, S assumes that R will understand the request for information and the privacy policy description. R makes few assumptions about the content of the capability request, and screens it carefully to ensure that it is recognized and valid; S may choose to authenticate the capability request data, and R may choose to check its authenticity.
3. Using information from step 2, R checks the request from S, including the privacy policy description and any authenticating or other provenance information against its own policies for release of capability information, resulting in an authorization decision whether or not to release the requested information. A decision to release information will depend on a degree of trust that the information will be used according to the policy declared by S for use of that information. R assembles a description of its capabilities and preferences, assuming that they can be understood by S, and sends them. The capability data might be encrypted to prevent disclosure to parties other than R or S. S makes few assumptions about the content of the capability description, and screens it carefully to ensure that it is recognized and valid; R may choose to authenticate the capability data, and A may choose to check its authenticity.
4. Using information from step 3, S analyzes the description of R's capabilities, and selects a form of data to be sent. The decision to authorize access to this information was made at step 2, so it may be that no further authorization is needed. Alternatively, S may need to re-check its policy to ensure that R is permitted to access the particular version of data that matches R's capabilities and preferences. Implicit in this authorization is a degree of trust that R will use the data in ways acceptable to S. The data might be encrypted to prevent disclosure to parties other than R or S. On receipt of the data, R may trust that the data is free of undesirable content (e.g. viruses, profanity), or may examine the content to ensure such freedom. If the data determined to be safe, R trusts that it does indeed correspond to the original request and use it accordingly.


Receiver
Sender
R has belief and trust in S providing required information
1. Request for information
S authenticates R, checks authorization, determines capability info is needed.
R authenticates S, checks privacy statement against local policy
2. Request capabilities, supply privacy policy
S constructs capability request with privacy policy statement
R trusts S privacy policy and sends capability information
3. Provide capabilities
S authenticates response, analyzes capability information
S sends requested


R trusts and uses data provided by S. R may or may not conform to trust expectations of S
4. Send requested data
information in appropriate format that matches R capabilities

This illustration is also available in PDF format.

This example shows how a simple transaction may be decomposed into a series of simple information transfers using the trust and policy interactions described previously. In doing this decomposition, the transaction appears to become rather complex, because many of the trust and policy based decisions that are made implicitly in the context of a protocol request have been made explicit. But, where normal protocol behaviour is acceptable, no additional logic or complexity over and above the protocol implementation is required. Apart from the content negotiation and privacy policy description, the above example transaction, with full authentication and content encryption of each transfer, could be completed using a standard HTTP-over-TLS protocol implementation.

3. Resource Description Framework (RDF) -

TOC&nbsp;

The Resource Description Framework (RDF)[22] is a W3C-defined framework for representing metadata and arbitrary information. It is based on a graph data model, for which an XML serialization is defined.

It is usually possible to use "home grown" XML for applications where RDF is applicable, but RDF offers the following distinguishing characteristics:

- A fundamentally simpler data model: a directed labelled graph.
- Consistent use of URIs to identify things and concepts being described.
- Open-ended extensibility, allowing new information to be added at any time. RDF fully leverages XML namespaces[18] to allow mixing and sharing of data defined for different applications. (See also Tim Berners-Lee's note on Evolvability.)
- A uniform framework for exchanging information between applications that goes well beyond what is provided by XML in terms of the level of interoperability achieved.
- A substantial and growing body of application information design that can be used directly by any RDF application.
- A growing body of tools for storing and manipulating RDF in standard relational databases.
- Formal semantics, and a well-founded notion of entailment.
- A growing body of tools capable of performing various kinds of processing on arbitrary RDF data. Of particular relevance to trust modelling, contract negotiation and security are tools that apply formal reasoning techniques to RDF information.
- A basis for ongoing ontology- and rule-language design and development. These developments incorporate many ideas that are relevant to trust-related topics such as contract formation.

The power and value of using RDF in a trust modelling framework is the integration and sharing of information between applications. For an isolated application, there may be little practical value in using RDF, but when several applications need to exchange information, the common representation thus provided is most valuable. I liken this to the effect that Internet Protocol (IP) had on the deployment of communication services in allowing disparate systems to communicate with each other and enabling the development and deployment of new communication services. RDF fulfils a similar role with respect to information services.

In this way, RDF can enable a "network effect", allowing a trust framework to be composed from simple tools and simple rules, accepting and generating information as RDF expressions.

Existing data formats can be viewed as defining two parts: syntax and semantics. Of these, the semantics is the most difficult to match to a different definition. By comparison, syntax is relatively easy to convert without loss. Thus it may be that the way to integrate diverse existing data format standards is to map their syntax into RDF while exactly preserving their semantics. For XML-based data formats, this has been easily achieved in some cases using XSLT[21] transformations.

4. Secure MIME (S/MIME) -

TOC&nbsp;

S/MIME[10] is an IETF standard for signing and encrypting MIME content, and providing other related security services.

It can be used to provide end-to-end security for arbitrary MIME content (e.g. textual or binary data), where "end-to-end" means that its security does not depend on trusting any intervening systems through which the data may pass. S/MIME secures a data object, and is not really suitable for securing streaming data like real-time voice or video.

4.1 Content and concepts

As its name suggests, S/MIME uses MIME-encapsulation [2], thus it can carry arbitrary (textual or binary) data over a text-only data transfer channel.

4.1.1 Signed data

Signed data contains:

- a payload
- a signature block

The payload is passed in the clear, and can be accessed as a MIME body part without processing the associated signature information.

The signature information contains:

- a crytographic digest of the payload, signed using a public key cryptographic algorithm (typically RSA or DSA),
- a copy of the public key paired with the private key used to sign the digest,
- a name that identifies the signer,
- a certificate or certificates that can be used to verify the supplied public key, and
- optional additional attributes that are also covered by the signature.

The message digest algorithm and public key signing algorithm may vary, and the S/MIME format contains information sufficient to identify the actual algorithms used.

The signature information can include multiple signatures applied to the same payload.

4.1.2 Encrypted (enveloped) data

Encrypted (enveloped) data has a single MIME body part containing:

- encryption control information
- encrypted payload data

The payload data is encrypted using a symmetric block cipher (e.g. triple-DES) and a randomly-generated key. The encryption key is then encrypted using a public key algorithm using the public key of the intended recipient. For a message with several recipients, this can be repeated using the public key for each intended recipient.

The encryption control information for a single recipient contains the following information:

- a recipient identifier, and

- the encrypted content-encryption key for that recipient.

4.2 Syntax

S/MIME message syntax is based on MIME [2] and ASN.1 [43].

MIME is used to encapsulate the S/MIME data, and in the case of signed data a MIME multipart structure is used to separate the payload and signature information.

ASN.1 is used to encode the signature information or the encryption control information.

4.3 Relationship to trust management

The importance of S/MIME in trust management is that it allows information to be passed over an untrusted channel with confidence that it will arrive unmodified or unseen by third parties. It also allows a recipient of such information to be confident of its origin, to the extent that they can be confident that only the claimed originator has access to the private key used to sign the message.

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, S/MIME provides mechanisms for authentication and encryption.

4.4 Related specifications

- OpenPGP[6] performs a similar function, but uses its own web-of-trust model for verifying keys.
- S/MIME depends on X.509[42] certificates for key verification.
- An alternative to the style of object security provided by S/MIME is channel security provided by protocols such as TLS[4] or SSL. These protocols are easier to deploy, but are generally regarded as less secure as they depend on trustworthiness of the intermediate servers used to apply and decode the security protocol.

---

5. Open PGP (OpenPGP) -

TOC&nbsp;

OpenPGP[6] is another IETF standard for signing and encrypting data objects, and providing other related security services. It can be used to provide end-to-end security for arbitrary data objects.

5.1 Content and concepts

Like S/MIME, it provides for signing and encryption of tranferred data. Technically, every PGP message constructed according to the standard is signed then encrypted, though some applications may omit the encryption stage.

PGP implementations also provide a framework for key management through a "web of trust" mechanism, though this is not part of the open PGP message format. Each PGP user maintains a "keyring" of public keys with associated trust values. Two important variations of trust are: (a) trust to introduce a new public key and corresponding associated identity, and (b) trust that a given key for the purposes of signing messages from an identified party.

5.1.1 Signed and encrypted data

Signed data contains a payload and a signature block, which are constructed using a PGP-specific binary format, consisting of a sequence of "packets". The signed message is optionally compressed, then encrypted and assembled with a "key material" packet.

Finally, the encrypted data may be "Ascii armored", which allows the data to be passed over a text data channel, possibly subject to whitespace and line break modifications, without loss or corruption of data.

The message digest algorithm and public key signing algorithm may vary, and the OpenPGP format contains information sufficient to identify the actual algorithms used.

The signature information can include multiple signatures applied to the same payload.

5.1.2 Cleartext signature framework

The OpenPGP specification also provides for associating a signature with some clear text.

This uses the "Ascii armor" labelling structure for separating and labelling the clear text, and combining that with signature information. Using this, the message content can be viewed using non-PGP-aware agents, even if its authenticity cannot be verified.

5.1.3 Web of trust

PGP, upon which openPGP is based, has a distinctive approach to building and conveying trust in public keys.

Compared with the X.509 hierarchical certificate chaining model (see section X.509 Public Key Certificates), PGP employs a more grassroots-based "web of trust", in which any keyholder can express degrees of trust in another. In his book "Applied Crytography" [47], Bruce Schneier puts it like this:

> *There are no key certification authorities; PGP instead supports a "web of trust". Every user generates and distributes his own public key. Users sign each other's public keys, creating an interconnected community of PGP users.*

5.2 Syntax

OpenPGP uses a binary message format combined with a Radix-64 ASCII encoding ("Ascii armor") for transferring messages over text data channels.

A framework for using MIME to convey OpenPGP-formatted messages is described in RFC 3156[12], "MIME Security with OpenPGP".

5.3 Relationship to trust management

The importance of OpenPGP in trust management is that it allows information to be passed over an untrusted channel with confidence that it will arrive unmodified or unseen by third parties. It also allows a recipient of such information to be confident of its origin, to the extent that they can be confident that only the claimed originator has access to the private key used to sign the message.

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, OpenPGP provides mechanisms for authentication and encryption.

5.4 Related specifications

- S/MIME[10] performs similar functions, but uses X.509 certificates and corresponding public key infrastructure for verifying keys.
- OpenPGP could be used with X.509[42] certificates for key verification, but no special infrastructure is provided by OpenPGP for dealing with X.509 certificates. In practice, there is probably little point as the relevant details are all worked out for S/MIME. It is interesting to compare PGP's unstructured "web-of-trust" model with X.509's hierarchical approach to trust distribution.
- RFC 3156[12], "MIME Security with OpenPGP", describes how to use

OpenPGP with ⊣ MIME[2], and introduces MIME content types "application/pgp-encrypted", "application/pgp-signature" and "application/pgp-keys" for this purpose.

- An alternative to the style of object security provided by OpenPGP is channel security provided by protocols such as ⊣ TLS[4] or SSL. These protocols are easier to deploy, but are generally regarded as less secure as they depend on trustworthiness of the intermediate servers used to apply and decode the security protocol.

6. XML digital signatures (XMLDSIG) - ⊣ TOC&nbsp;

⊣ XMLDSIG[13] (also ⊣ [24]), or XML Signatures, is a joint IETF/W3C standard defining XML Signatures that can be used for signing arbitrary data, but with a particular view to XML content.

It is similar to the signing capabilities of S/MIME and OpenPGP in that it provides object authenticity and integrity protection.

Unlike S/MIME and OpenPGP, XML Signatures can exploit the internal structure of XML (or other data formats) to provide for selective signing and canonicalization of equivalent representations, allowing a message containing content thus signed to be manipulated (within certain bounds) without invalidating the signature.

6.1 Content and concepts

⊣ XMLDSIG[13] provides for signing (only) of XML data. This provides for authentication and integrity protection of XML-formatted objects.

XML Signatures are applied to arbitrary digital content (data objects) via an indirection. Data objects are digested, the resulting value is placed in an element (with other information) and that element is then digested and cryptographically signed. XML digital signatures are represented by a <signature> element (qualified with the XML digital signature namespace).

The signature is linked to the signed data by a URI reference.

Signed XML data can be embedded in the same document as the XML signature, and referenced by a URI reference consisting of just a fragment identifier. Such local data can be included within an "enveloping signature" or can enclose an "enveloped signature". "Detached signatures" are over external network resources or local data objects that reside within the same XML document as sibling elements; in the latter case, the signature is neither enveloping (signature is parent) nor enveloped (signature is child).

An XML signature is constructed in 3 parts:

1. Each target data element to be covered by the signature is associated with a <reference> element that indicates:
   - one or more transformation algorithms to be applied, which may include algorothms for canonicalization,
   - a digest algorithm algorithm that is calculated over the transformed content, and
   - the digest value that must match the computed digest value.
2. A signature over the target data description, including the digest values to be matched, consisting of:
   - A canonicalization method that is applied to the target value descriptions,
   - a signature method used to verify the target data description, and
   - a signature value calculated using the indicated signing method applied to the canonicalized target data description (including the canonicalization and signature method indicators).
3. Key information about the signing key used, which can indicate certificate information or some other kind of key identification.

Unlike the signing mechanisms of S/MIME and OpenPGP, there is a level of indirection here between the signature itself, and the data that is signed. The assured relationship between the data and the signature is maintained by the digest value(s) that are part of the indirection data. This indirection allows a signature value to be calculated over non-continguous data, and even a composition of data from several different sources. It also allows for the application of transformations to make the signature less sensitive to minor format variations.

6.2 Syntax

The syntax of an XML Signature is entirely based on XML.

Through the indirect data description element (which is also XML) the information signed can be any combination of XML and other, non-XML, data formats.

6.3 Relationship to trust management

The importance of XML Signature in trust management is that it allows information to be passed over an untrusted channel with confidence that it will arrive unmodified by third parties, and allows a recipient of such information to be confident of its origin, to the extent that they can be confident that only the claimed originator has access to the private key used to create the signature.

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, XML Signature provides a mechanism for authentication.

6.4 Related specifications

- The XML Signature specification is very much built upon XML[15] and XML namespaces[18].
- XML canonicalization[19] is used as an XML canonicalization algorithm, for which support is required.
- The Base64 algorithm of MIME[2] is used for transforming binary data to textual format for the purposes of signature calculation.
- XML canonicalization[20] is used to select (filter) parts of an XML document for signing.
- XML transformations[21] may be used to apply arbitrary transformations to an XML document for signing.
- XML signatures can be used in conjunction with X.509[42] certificates for key verification. Several other forms of key information are also supported.
- XKMS[26] is likely to be a useful key verification mechanism for use with XML signatures.
- S/MIME[10] also performs data object authentication functions, but with many fewer options and less flexibility concerning what is signed.
- OpenPGP[6] also performs data object authentication functions, but uses its own web-of-trust model for verifying keys.
- XML Encryption[25] is a counterpart to XML signature, and provides for protecting the privacy of content in XML messages.
- An alternative to the style of object security provided by XMLDSIG is channel security provided by protocols such as TLS[4] or SSL. These protocols are easier to deploy, but are generally regarded as less secure as they depend on trustworthiness of the intermediate servers used to apply and decode the security protocol.

6.5 Note

Because of its great flexibility and relative complexity, great care should be used when applying XML signatures.

In particular, consideration should be given that the selection of data elements to be signed, and application of transformation rules, may result in the signed data conveying some different information than intended by the user who allowed their key to be used to make the signature.

It may be that it is important to know what application created an XML signature, so that trust in the signature value can be judged accordingly. (To some extent, this applies to any form of digital signature, but the wide range of XML signature options leaves greater possibility for unintended consequences.)

Where an XSLT transform is used to transform the content, it should either be served by a trusted source, or a digest of the actual stylesheet used included as part of the signature.

7. XML encryption (XMLENC) -

⊣ TOC&nbsp;

⊣ XMLENC[25] is a W3C standard for encrypting arbitrary data and representing the result as XML. The result of encrypting data is an XML Encryption element which contains or references the cipher data.

7.1 Content and concepts

An encrypted data element may contain:

- details of the encryption method used,
- information about the encryption key used,
- a copy of or reference to the encrypted data, and
- an EncryptionProperties element, which can contain additional information concerning the generation of the encrypted data element (e.g., date/time stamp).

When used to encrypt XML data, the encryption may be applied selectively. That is, selected elements in the clear form of the document may be replaced by an encrypted data element.

7.2 Syntax

The syntax of an XML encrypted data element is entirely based on XML.

Through the mechanisms of indirect reference, the encrypted data may be passed as XML or some other, non-XML, data format.

7.3 Relationship to trust management

The importance of XML Encryption in trust management is that it allows information to be passed over an untrusted channel with confidence that it will arrive unseen by third parties. This capability is critically important for passing private authorization information as part of a transaction.

In relation to the message transfer framework described in section ⊣ Interactions between autonomous communicating parties, XML Encryption provides a mechanism for encryption.

7.4 Related specifications

XML Encryption builds in many ways on the XML Signature specification, and consequently bears some relation to several of the same specifications (see section ⊣ XML digital signatures (XMLDSIG), sub-section ⊣ Related specifications, for more details).

8. X.509 Public Key Certificates -

⊣ TOC&nbsp;

⊣ X.509[42] is an ITU standard format for public key certificates.

Public key certificates are a key element in the distribution and transfer of trust in public keys, which is itself is the basis for any other transfers of trust that depend upon public key cryptography.

8.1 Content and concepts

The purpose of a public key certificate is to distribute public key information in a secure, well-managed fashion.

Public key cryptography depends critically on the user of a public key having fell-founded confidence that the corresponding private key is known only to the person that they believe owns it. So, when checking a signature, the public key used must correspond to the intended signer's private key. Similarly, when encrypting data for a given recipient, the public key used must correspond to a private key that is known only to the intended recipient.

X.509 is based around the concept of a Certifying Authority (or CA) who checks the identity of some person or authorized entity who has also proved that they have posession of the private key. The CA then issues a Certificate, which is a data structure containing (among other things) the key holder's identity and a copy of their public key, all signed using the CA's private key. Then, if the recipient trusts the CA to properly identify the person, they can also trust that the identified person can also create signatures that check out using the public key signed by the CA.

An important property of a Public Key Certificate is that it can be made publicly available through untrusted channels without thereby compromising any trust that may be vested in the key.

8.1.1 Certificate limitations and revocation

The use of certificates is fine for checking that the intended signer did indeed have posession of the private signing key, but can never prove that nobody else also had access to the same key. This is an inherrent problem with any PKI-based trust system. The relying party must trust that the signer's private key has not been compromised by disclosure to any other party.

A partial response to the problem of signing key disclosure to unauthorized parties is the Certificate Revocation List (CRL), which is a published list of certificates that have been revoked. Another partial response to this problem is that an X.509 certificate has a limited lifetime, after which it cannot be regarded as valid, and a new certificate must be obtained.

Yet another response to the problem is for the relying party to always check the certificate with the CA when using it to verify a signature. This may be expensive in terms of computation cost and communication delays, but is often regarded as the only acceptably safe option for high-value transactions.

Thus, one can see that trust in a public key depends on a number of factors, including trust in the key holder, both to live up to obligations conveyed by their signature, and also to properly guard their private key from disclosure to unauthorized parties, trust in the CA to properly verify the key holder's identity, and trust in the security of the technology used to create and verify signatures.

8.1.2 Certificate chaining

As described above, there is an implicit assumption that the user of a public key will trust the CA that was used to sign the corresponding certificate. This is not a practically scalable proposition.

So X.509 employs the idea of certificate chains, where each CA's public key is itself signed by a "higher" CA, and so on until a trusted "root" CA is encountered. Thus, a chain of certificates can link the holder of some key with a user of the corresponding public key to a common point of trust. Set against this, the longer the certificate chain the more scope there is for compromise of any one of the CA signing keys, which would effectively nullify the basis for trust in the end user keys thus protected.

The original X.509 design called for a single trusted root to the CA hierarchy. In practice, that has not been realized, but there are a number of certificate issuing organizations that are widely recognized (if not trusted), and whose public keys are widely distributed with web browsers and other software.

An unfortunate aspect of many kinds of computer interaction is that ordinary users don't have the real-world clues upon which to base their trust decisions ("would you buy a used car from this man?"). For the most part, non- technical computer users (and many technical ones too) have insufficient knowledge and understanding to make reasonable trust decisions, and end up relying on personal recommendations and credit card guarantees. The former is alright to a point, but is not fully applicable at Internet scale. The latter effectively puts credit card companies in the position of trusted certifying authorities for certain kinds of purchases, but doesn't extend so easily to other kinds of transaction (e.g. online medical advice).

Thus, public key certificates in general, and X.509 in particular, are technical approaches to a problem that cannot be solved wholly by technical means. X.509 certificates can carry additional information about policies and other non-technical aspects of trust decisions.

I expect that X.509 has a part to play, but it cannot be regarded as a panacea for all trust problems.

8.2 Syntax

X.509 certificates are defined using ASN.1, per X.680[43], and encoded for transmission using Basic Encoding Rules (BER) per X.690[45]

8.3 Relationship to trust management

Public key certificates are a technical mechanism for conveying trust in (the authenticity of) public keys.

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, public key certificates support the use of public key cryptography to provide authentication (by confirming the correct public key to verify a signature) and encryption (by confirming the correct public key to use in encrypting a message).

8.4 Related specifications

- S/MIME[10] is designed to be used with X.509 certificates, and has explicit provision for carrying X.509 certificates as part of the signing and encryption key information.
- X.509 certificates can be used to certify public keys for use with OpenPGP[6] as an alternative or in addition to the PGP "web of trust", specific mechanisms for doing this are not widely deployed [[[I think - is this true?]]].
- X.509 certificates are quite complex, and their use involves the construction and analysius of ASN.1 BER-encoded data. This can be a significant barrier to implementation, and the XKMS[26] protocol is being developed to provide an easier way for XML applications to access the facilities provided by X.509 and other certificates.

9. Internet X.509 Public Key Infrastructure (PKIX) -

TOC&nbs p;

PKIX[14] is an IETF standard profile of X.509 for Internet use.

For further information, see section X.509 Public Key Certificates and the PKIX specification document [14].

10. XML Key Management Services -

TOC&nbs p;

XKMS[26] is W3C protocol specification for verifying and accessing information about X.509 public key certificates (or other forms).

10.1 Content and concepts

XKMS[26] specifies protocols for distributing and registering public keys, suitable for use in conjunction with the proposed standard for XML Signature [24] and as an anticipated companion standard for XML encryption[25].

The XML Key Management Specification (XKMS) comprises two parts:

1. the XML Key Information Service Specification (X-KISS), and
2. the XML Key Registration Service Specification (X-KRSS).

The X-KISS specification defines a protocol for a Trust service that resolves public key information contained in XML-SIG elements. The X-KISS protocol allows a client of such a service to delegate part or all of the tasks required to process XML Signature key information elements. A key objective of the protocol design is to minimize the complexity of application implementations by allowing them to become clients of a key information and registration service, and thereby to be shielded from the complexity and syntax of the underlying PKI used to establish trust relationships. The underlying PKI may be based upon a variety of different specifications such as X.509/PKIX, SPKI or PGP.

The X-KRSS specification defines a protocol for a web service that accepts registration of public key information. Once registered, the public key may be used in conjunction with other web services including X-KISS.

10.2 Syntax

The syntax of an XKMS is entirely based on XML.

10.3 Relationship to trust management

XKMS is a protocol that provides access to the facilities conveyed by X.509 certificates, as well as other forms of public key certificate. As such the relation to trust management is largely the same as for X.509: see section X.509 Public Key Certificates for details.

10.4 Related specifications

- XKMS can be used in conjunction with X.509[42] certificates, or some other form of public key information.
- The XKMS specification is very much built upon XML[15] and XML namespaces[18].
- XKMS is likely to be used in conjunction with XML Signature[24] and XML Encryption[25].

---

11. Kerberos ticket issuing systems -

TOC&nbsp;

Kerberos[1] is a widely used trusted third party authentication technology developed originally at MIT for Project Athena.

11.1 Content and concepts

Kerberos is an authentication protocol that depends for its security on shared secrets with a trusted third party, and employs symmetric cryptography.

The components in the Kerberos model are

- A Kerberos server
- Ticket-granting servers
- Application servers
- Clients

To use an application service, a client must obtain a ticket for that service, from a ticket granting server. But to obtain a ticket, the client must obtain a ticket-granting-ticket from the Kerberos authentication server.

Clients each share a secret key with the authentication server (AS), which is used to encrypt session keys generated by AS. AS also shares a secret key with the ticket granting server(s) (TGS), which is used to encrypt the ticket issued, so only TGS can access the content of the ticket to create service tickets.

Ticket granting servers similarly share secrets with the services for which they issue tickets. This, in conjunction with a shared session key issued and encrypted by AS, allows TGS to create a service ticket for the client to present to gain access to a service.

There are many additional details in the design of Kerberos, but the basic idea can be seen to be propagation of trust in identity through a chain of chared secrets.

11.2 Syntax

The Kerberos protocol syntax is defined using ASN.1, and encoded for transmission using the Distinguished Encoding Representation (DER) of the data elements as described in the X.509 specification, section 8.7 [42].

(The above citation was copied from from RFC 1510, but doesn't agree with the X.509 (1994) document itself. See X.690[44] for the current decsription of ASN.1 distinguished encoring rules (DER).)

The kerberos specification allows for the ASN.1/DER encoded data to be encapsulated in application protocol frames if it cannot be carried directly.

11.3 Relationship to trust management

Kerberos provides two important capabilities: authentication of network entities, and distribution of session keys for encryption of data.

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, Kerberos provides authentication and support for client-to-service encryption by securely distributing session keys.

Note that Kerberos is oriented toward supporting client access to some set of services, rather than arbitrary person-to-person (or agent-to-agent) security requirements.

11.4 Related specifications

- X.509[42] provides similar capabilities based on public key cryptography, and a hierarchy of certifying authorities.

---

12. Security Assertions Markup Language (SAML) -

TOC&nbsp;

SAML[37] is an OASIS standard format and protocol binding framework for conveying security assertions. A particular goal of SAML is support for single sign-on, a single user authentication being used to generate sufficient credentials to access resources in different domains.

12.1 Content and concepts

The Security Assertion Markup Language (SAML) is an XML-based framework for exchanging security information. This security information is expressed in the form of assertions about subjects, where a subject is an entity (either human or computer) that has an identity in some security domain. A typical example of a subject is a person, identified by his or her email address in a particular Internet DNS domain.

Assertions can convey information about authentication acts performed by subjects, attributes of subjects, and authorization decisions about whether subjects are allowed to access certain resources.

12.1.1 Assertions

An assertion is a package of information that supplies one or more statements made by an issuer. SAML allows issuers to make three different kinds of assertion statement:

- Authentication: the specified subject was authenticated by a particular means at a particular time.
- Authorization decision: a request to allow the specified subject to access the specified resource has been granted or denied.
- Attribute: the specified subject is associated with the supplied attributes.

Specific XML elements are defined for introducing additional kinds of assertion into a SAML expression.

Assertions have a nested structure. A series of inner elements representing authentication statements, authorization decision statements, and attribute statements contain the specifics, while an outer generic assertion element provides information that is common to all of the statements; e.g. issuer, time of issue, assertion qualifiers, etc.

12.1.2 Protocol

The security assertion language definition is free-standing, and assertions can be conveyed in a number of different ways, using any protcol capable of delivering XML content.

SAML also defines a simple request-response protocol, for discovering information about SAML assertions held by some authority.

The protocol defines a number of query types, and also provides an extension point element for introducing new queries.

The response to a query is a status value and zero or more SAML assertions.

The SAML protocol may be bound to a variety of data transfer protocols. Once such specific binding is defined by the SAML specification: SOAP over HTTP [34] [35] [8].

12.2 Syntax

The assertion language uses XML[15], and is described using XML schema[16].

12.3 Relationship to trust management

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, SAML provides a format for describing authentication, authorization and other information that may be derived from or input to a policy, and as such may be used to impact any policy-based decision.

12.4 Related specifications

- SAML appears to provide some of the functions of certificates [42] and key management services [26], but not covering a complete specification of how the information is secured. It also provides additional information that is not generally provided by certificates, but may result from the use of certificates and associated security technologies.

- The relationship to policy-based decisions suggests that SAML could usefully be used in conjunction with XACML[38].

13. Extensible Access Control Markup Language (XACML) -

TOC&nbsp;

XACML[38] is an OASIS specification for using XML to expressing policies for information access over the Internet.

XACML is designed to address fine grained control of authorized activities. It takes account of the effect of characteristics of the access requester, the protocol over which the request is made, authorization based on classes of activities, and content introspection (i.e. authorization based on both the requestor and potentially attribute values within the target where the values of the attributes may not be known to the policy writer).

XACML also aims to suggest a policy authorization model to guide implementers of authorization mechanisms.

13.1 Content and concepts

A policy is a rule or rules of behaviour (or rules that affect behaviour of a system in some way).

The complete policy applicable to a particular decision request may be composed of a number of individual rules or policies. For instance, in a personal privacy application, the owner of the personal information may define certain aspects of disclosure policy, whereas the enterprise that holds the information may define certain other aspects. In order to render an authorization decision, it must be possible to combine the separate policies.

Many of the ideas in XACML seem to derive from work by Sloman [48].

13.1.1 Rules, policies and policy sets

XACML uses three basic concepts for constructing policies:

- Rule: a simple Boolean expression that can be evaluated based on supplied information. A rule is evaluated with respect to some target set of resources, requesting subjects and requested actions, and its evaluation as Boolean true may result in a Permit or Deny result. If the expression evaluates as false, then the output from the rule is indeterminate (not specified). Thus, the result of evaluating a single rule is: Permit, Deny or Indeterminate.
- Policy: a set of rules, together with a specified procedure for combining the results of their evaluation. The result of evaluating a single policy is: Permit, Deny or Indeterminate.
- Policy set: a set of policy and policy set elements, together with a specified procedure for combining the results of their evaluation. The result of evaluating a policy set is: Permit, Deny or Indeterminate.

If a target (i.e. resource, requesting subject and requested action dewtails) is not specified as part of a rule, it is inherited from a policy that invokes the rule.

13.1.2 Combining results

The following methods are defined for combining the results of rule, policy and policy set evaluation:

- deny-overrides,
- permit-overrides,
- first applicable, and
- only-one-applicable-policy.

13.1.3 Decision requests and XACML contexts

XACML defines a common abstraction, or canonical form, for policy decision requests, to which all requests for an authorization decision can be mapped. This is called an XACML context.

13.1.4 Obligations

In some cases, access to a resource may be associated with some obligations. Enforcement of such obligations may depend on the network entity that enforces the access policy. XACML provides a framework for specifying any such obligations.

13.2 Syntax

The access control language uses XML[15], and is described using XML schema[16].

13.3 Relationship to trust management

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, XACML appears to address areas of policy-based decision making concerned with access control (i.e. decisions to permit or deny access to perform some action on a resource).

13.4 Related specifications

- SAML[37] is a specification that can provide raw information upon which XACML access decisions can be based, and be used to represent the result of such decisions.
- A completely different, and possibly much simpler, approach to the rule evaluation problems addressed by XACML would be to use RDF[22] and RDF-rule and inference languages such as DAML+OIL[36].

---

14. Web Services Security (WSS) -

TOC&nbsp;

WSS[39] is an OASIS specification for enhancements to the SOAP[34] messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication.

The specification provides three main mechanisms:

1. ability to send security token as part of a message,
2. message integrity, and
3. message confidentiality.

These mechanisms by themselves do not provide a complete security solution for Web services. Instead, they are a building block that can be used in conjunction with other Web service extensions and higher-level application-specific protocols to accommodate a wide variety of security models and security technologies.

The WSS specification contains a message security model, which notes that a "signature can be used to 'bind' or 'associate' the message with the claims in the security token (assuming the token is trusted)". Thus, it seems there is a clear interface indicated here between security mechanisms and trust.

At the time of writing, the WSS specification is still a work-in-progress, so changes and additions are to be expected.

14.1 Content and concepts

14.1.1 Message security model

WSS defines a message security model, that is described in terms of security tokens (claims), endorsement of claims (e.g. by digital signature) and verifiable proof of posession. See section 3.1 of the specification.

14.1.2 SOAP security header

WSS defines a SOAP security header blocks, for attaching to a message security related information targeted at a specific receiver.

The core of SOAP[34] is an envelope framework for XML messages. This framework assumes that a message is passed from a sender to an ultimate receiver by way of zero, onw or more intermediate receivers. All receivers are identified by URIs (which may be specific to a receiver, or indicative of a role performed), and a SOAP envelope may contain any number of header blocks targetted to any receiver so identified.

A single message may have multiple security header blocks provided thay are targetted to different receivers.

14.1.3 Global identifier attribute

WSS defines a global identifier attribute as a preferred and simple way to identify specific XML content in a SOAP message.

14.1.4 Security tokens

WSS describes a number of specific security tokens that may appear in a security header block, corresponding to some common kinds of security claims:

- Username and password.
- Binary tokens, such as X.509 certificates or Kerberos tickets.
- XML tokens, such as XML signatures (but not limited ti such).

The WSS specification also discusses mechanisms for referencing security tokens.

14.1.5 Signing messages using XML Signature

WSS defines a profile of XML signature[24] for use when signing SOAP messages.

This profile specifies a numbert of options that are left open by the XML signature specification:

- XML exclusive canonicalization http://www.w3.org/2001/10/xml-exc-c14n#.
- XML decryption transformation http://www.w3.org/2001/04/decrypt#.
- How to include a signature within a security header block.
- How to validate a signature within a security header block.

14.1.6 Encrypting messages using XML Encryption

WSS defines a profile of XML Encryption[25] for use when encrypting SOAP messages.

14.1.7 Message timestamps

WSS defines a a number of message elements to be used for attaching various kinds oif timestamp information to a message; e.g. expiration, creation, tracing.

14.2 Syntax

WSS uses XML[15], and is described in part using XML schema[16].

It is for use in the context of SOAP[34] messages.

14.3 Relationship to trust management

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, WSS deals with a number of aspects of integrating mechanisms for authentication, privacy and authorization into an application framework based on SOAP. In particular, the message security model makes quite explicit a number of areas where message security mecahnisms must interface with trust assumptions or evaluations.

14.4 Related specifications

- The WSS specification is very much built upon SOAP[34], which is in turn build upon XML[15] and XML namespaces[18].
- The WSS specification is intended to be used with XML signature[24], XML Encryption[25], and other security mechanisms. See their related specifications in this document for further information.
- Security tokens may come in a variety of forms, including: X.509 certificates[42], SAML assertions[37], etc.
- XKMS[26] may be a useful mechanism for verifying security assertions that are not completely self-verifying, such as signatures without certificates.

15. Platform for Internet Content Selection (PICS) -

TOC&nbsp;

PICS [27][28][29][30] is W3C specification for enabling Web users to select the kind of content they wish to receive. Web browser software can read service descriptions in order to interpret content labels and assist end-users in configuring software to select delivery of desired content.

Labels are provided by a "rating service", according to a "rating system".

PICS cannot, of itself, guarantee that the content is in fact in accordance with the rating labels given. Labels assigned are entirely in the gift of a rating service, and reliability of the labels is a matter for a user's trust in the competence and integrity of the rating service.

The PICS specification contains 4 parts:

1. a definition of the PICS rating service and rating system description syntax [27].
2. a definition of the PICS rating labels, and mechanisms for distributing label values [28].
3. a language for writing profiles, which are filtering rules that allow or block access to URLs based on PICS labels that describe those URLs [29].
4. a method of adding extensions to PICS 1.1 labels for purposes of signing them using XML digital signatures [30].

15.1 Content and concepts

15.1.1 Rating service

A "rating service" is provided by an individual or group who examine the content of Internet resources and provide labels to indicate the nature of that content.

PICS is content neutral, and does not define the rating labels: these are provided by a "rating system".

15.1.2 Rating system

A rating system specifies the dimensions used for labeling, the range of allowable values on each dimension, and a description of the criteria used in assigning values. For example, the MPAA rates movies in the USA based on a single dimension with allowable values G, PG, PG-13, R, and NC-17.

Each rating system is identified by a valid URL. This enables several services to use the same rating system and refer to it by its identifier. The URL naming a rating system can be accessed to obtain a human-readable description of the rating system. The format of that description is not specified.

15.1.3 Content label

A content label (or rating) contains information about a document.

A content label (or rating) has three parts:

- The URL naming the rating service that produced the label,
- A set of attribute-value pairs, which provide information about the rating such as the date that the rating was assigned. Some such labels are defined by the PICS specification.
- A set of rating-system-defined attribute-value pairs, which actually rate the item along various dimensions (also called categories).

15.2 Syntax

PICS uses a LISP-like textual syntax for service and rating system descriptions. A MIME content-type, application/pics-service, is defined for labelling PICS service description content (which also contains rating system descriptions).

PICS also uses a textual syntax for content labels, and a MIME content-type application/pics-labels to contain labels when distributed in isolation from the content they label.

15.3 Relationship to trust management

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, PICS is primarily concerned with "exposure control". PICS rules describe aspects of policy associated with exposure control.

There are some ancillary trust interactions to be considered when using PICS:

- Does the party whose exposure to undesirable content is being managed trust the rating service that provides the content labels? Answering this question is necessarily a wider issue that cannot be resolved by PICS alone.
- Is the rating information provided truly what the rating service applied to the corresponding content?
- Is the content provided a copy of the content actually reviewed by the rating service?

the last two issues are addressed (subject to the user's ability to verify a signature of the rating service) by the PICS signed label extension [30].

15.4 Related specifications

- PICS was an early foray into the area of defining metadata for Web resources. The metadata ideas were later picked up and generalized by the RDF effort [22] [23]. However, PICS does more than just define a metadata format, and even when using RDF to describe such information, the application design lessons from PICS have a part to play.
- The PICS signed label specification is closely allied to, and depends upom, the XML digital signature specification [24].
- The selection rules element of PICS seems to address some of the goals that later appear in XACML [38]. The PICSRules are, however, very much easier to understand, though they probably don't have the ultimate flexibility

of XACML.

16. Platform for Privacy Preferences (P3P) -

TOC&nbsp;

P3P[31] is a W3C specification that enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P-enabled user agents will allow users to be informed of site practices and automation of decisions to release private information based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit.

16.1 Content and concepts

In order to deliver personalized or adapted content, web sites sometimes require that browsers provide information about the user. Some of this information may considered to be sensitive personal information that the user does not wish to be widely disseminated, or used for targetting of information and/or services.

Different web publishing organizations have different policies with respect to personal information: a user may be prepared to release personal information to some but not others.

P3P version 1.0 is a protocol designed to inform Web users of the data-collection practices of Web sites. It provides a way for a Web site to encode its data-collection and data-use practices in a machine-readable XML format known as a P3P policy.

The P3P specification defines:

- A standard schema for data a Web site may wish to collect, known as the "P3P base data schema".
- A standard set of uses, recipients, data categories, and other privacy disclosures
- An XML format for expressing a privacy policy
- A means of associating privacy policies with Web pages or sites, and cookies.
- A mechanism for transporting P3P policies over HTTP.

16.1.1 Policy reference file

A policy reference file links one or more policy statements with different parts of a web site, indicating the relevant data collection and use policies for the different areas. It also contains information about the durability of the policy references, so that a user agent can make sensible policy caching arrangements. (Note that individual policy files also carry their own expiry information.)

The policy reference file is a starting point from which all P3P policy information for a site can be traced.

16.1.2 Policy file

A policy file defines actual data collection and usage policies for any URIs with which it is associated.

It may encode the following information:

- Reference to human-readable version of data collection and usage policy.
- A precise description of the legal entity making the representation of the privacy practices.
- Means for access, if any, to identified collected data.
- Description of or reference to dispute resolution procedures that may be followed for disputes about a services' privacy practices.
- Possible remedies in case a policy breach occurs.
- Identification of information that may be collected related to use of the site.
- Purposes of data collection, or uses of the collected data.

- Recipients to whom the collected data may be disclosed.
- Retention policy for collected data.

16.1.3 Statements and data groups

In addition to some general information, a P3P policy contains one or more statements, each of which describes data practices that are applied to particular types of data.

A "data group" element identifies the kinds of data to which the data practices apply, and contains a number of "data" elements.

A "data" element identifies some simple or composite item of data that may be collected. The data is identified by a URI reference that has the form 'URI#name', (which may be constructed from a separately specified base URI and relative URI reference):

- The 'URI' part is considered to identify a data schema, which defines a hierarchically arranged set of data items.
- The '#fragment' is considered to identitify an item in the data schema, and uses dots ('.') to separate names of elements in the hierarchy. A data item with a name of the form '#xxx' is presumed in include all data items with names of the form '#xxx.yyy'.

16.1.4 Base data schema

P3P defines a base data schema, which is the default data schema used if no other data schema URI is indicated by a data group of data element.

All P3P-compliant user agent implementations are requred to be aware of and understand the data elements in the P3P base data schema. This provides a basic guarantee of interoperability for P3P systems.

The base data scheme employs some basic data structures (i.e. common hierarchical arrangements of values, such as a date consisting of day, month and year). It is recommended that these basic data types are reused where possible by other data schemas.

16.2 Syntax

The P3P policy reference and policy files use XML[15], and their syntax is normatively defined using XML schema[16].

16.3 Relationship to trust management

In relation to the message transfer framework described in section Interactions between autonomous communicating parties, P3P is concerned with privacy, through mechanisms of policy description, negotiation and trust.

Note that, in the final analysis, a user must trust a web site controller to behave according to its indicated policy and use information only in the ways indicated. If data is retained for extended periods of time, the user must also trust that future management of the organization will continue to apply the agreed policies.

There has been some legal activity associated with P3P. It is not currently clear to what extent a declared P3P policy can be considered to be legally binding on a web site controller.

[[[This is a question to ask of the iTrust WG.]]]

16.4 Related specifications

- P3P semantics are a natural candidate for expression in RDF [22] [23]. Indeed, some work has been done to define an RDF schema for P3P [32].
- P3P has a natural counterpart in CC/PP[33]. CC/PP defines an RDF-based format for providing client capabilities and user preferences to a data provider. A P3P policy would naturally be used in determining what

information to release in a CC/PP profile. Unfortunately, CC/PP uses arbitrary URIs to identify attributes in a profile, where P3P uses a restricted form, so it's not necessarily trivially easy for a P3P profile to describe its policy with respect to CC/PP attribute data.
- It is possible to consider using a framework like PICS [27] [28] to rate the reliability of a web site's privacy profile, which can be treated as web content like any other. This could be an important step to establishing a required degree of trust in a web site's behaviour with respect to private information.

17. Composite Capabilities/Preference Profile (CC/PP) - TOC&nbsp;

CC/PP[33] is a W3C specification for describing user agent capabilities and user preferences so that content may be suitably selected or adapted.

CC/PP is an application of RDF [22] [23], which uses the RDF framework to convey assertions about the client capabilities and user preferences.

A CC/PP profile is delivered to a data provider which uses the information thus provided to select or construct a data format most suited to the indicated capabilities and preferences.

17.1 Content and concepts

A CC/PP profile is organized as a 2-level hierarchy, with the first level being components, corresponding to broad areas of user agent functionality, and the second level being attributes, curresponding to detailed capabilities or preferences related to the containing component.

The 2-level hierarchy is somewhat arbitrary, but seems to be a reasonably good fit for many practical requirements.

17.1.1 Components

A component corresponds to a major area of user agent functionality, such as:

- Hardware platform
- Software platform
- Browser user interface
- Network characteristics
- etc.

(This part of the design comes straight from UAPROF UAProf[49].)

Each type of component is associated with an RDF class. An RDF schema [23] can be used to describe the properties that are used with a given CC/PP component type.

17.1.2 Attributes

CC/PP attributes describe specific user agent features associated with a CC/PP component. They are represented as RDF properties, and each attribute is therefore identified by a URI [5].

17.1.3 Defaults

Default descriptions for particular kinds of devices can be constructed as separate documents and referenced (using a URI) by a CC/PP profile. In this way, many features which are common to a range of such devices can be stored separately at a generally accessible location and do not need to be transferred with every client request for data. A client profile can explicitly specify any attributes which are different from a default profile thus referenced.

This is particularly important for mobile devices, where bandwidth available to the client may be very limited.

It also provides a very simple basis for creating distributed profile definitions in some limited ways. There has been some consideration of more complex mechanisms for combining separately defined elements into a profile, but these are not part of the current CC/PP work.

17.1.4 Proxy chaining

An optional feature of CC/PP is support for proxies in a request chain that may have the effect of modifying the range of data that a data provider may send.

The introduction of proxies introduces some additional trust considerations, and the design of CC/PP allows that the source of profile information can be preserved so that appropriate trust related decisions can be made by the data provider.

17.2 Syntax

CC/PP is an application of RDF, and uses the RDF/XML syntax ⊣ [22].

17.3 Relationship to trust management

In relation to the message transfer framework described in section ⊣ Interactions between autonomous communicating parties, CC/PP deals primarily with achieving dependability, by transferring information that improves the chances of delivering usable information.

CC/PP in turn depends on a degree of trust that sensitive personal information supplied in a profile will not be mis-used, and that the information will be used to improve the usability of data supplied rather than to degrade it.

When proxies are involved, it is necessary that they be trusted to convey the profile faithfully, and also to faithfully convey the selected data, applying transformations only if those are expected and/or authorized by the receiving client.

17.4 Related specifications

- CC/PP has it's origins in ⊣ UAProf[49], a specification of the WAP forum. It has also been influenced by the IETF CONNEG work ⊣ [11]⊣ [7].
- The most obvious related specification does not yet exist, namely one to describe source data characteristics in a way that can be matched with a CC/PP profile. There has been some mention that XHTML modules might have a part to play in this respect, but I am unaware of any generic effort to match these with CC/PP profiles.
- CC/PP has a natural affinity with P3P, which is discussed in section ⊣ Platform for Privacy Preferences (P3P) above.
- Some work on content negotiation was performed by the IETF CONNEG working group ⊣ [11]⊣ [7]. Although this work does not use RDF, it is in many ways closer to the spirit of RDF, combining assertions about various network resourcesto reach conclusions, than CC/PP which takes a very partial view of the overall content negotiation and content adaptation problem.

---

18. Security considerations -

⊣ TOC&nbsp;

I adopt the following working definition of "trust":

> *The firm expectation that an entity will behave dependably and securely within a specified context.*

The whole idea of trust management is intimately bound with security. Most security technologies convey trust in some way, though these alone cannot establish or create trust.

Much of the standards work conducted to date has focused on security mechanisms, and have left the establishment of trust as a "problem for the reader". The difficulty of establishing and describing trust underpins the failure to achieve meaningful large-scale deployment of a public key infrastructure: by simply focusing on the technology to transfer trusted assertions, much needed parts of the larger picture have been left unaddressed.

In focusing on the establishment of trust, I believe it is important that we do not forget that the mechanisms for conveying trust decisions must be reliable: the work that has gone into secure systems design, and the resulting designs and standards, are highly relevant to any work in trust management.

---

19. Conclusions -

⊣ TOC&nbs p;

There are many specifications that address some aspects or other of Internet security, and which all relate to trust in some way. Some depend on trust for their effectiveness, some can be used to convey trusted information from one place to another, or automate trust-related decisions from supplied data according to specified policies.

In this, trust is a means to an end, rather than an end in itself. Trust is an attitude, as distinct from some piece of conclusive evidence, that allows a user to permit some action to be taken on their behalf.

The specifications that have been surveyed here work together to a greater or lesser extent. In the areas most closely related to trust, it seems lesser rather than greater.

Many of the specifications most closely related to trust concerns are based on XML (P3P, SAML, XACML, etc.) Unfortunately, while XML provides a common syntactic framework it doesn't guarantee that the information conforming to the different specifications is easily combined. In some cases, the specifications use common elements which improve interworking between them.

One of the hopes for RDF, which the SWAD Europe project seems well set to explore, is that it may be used as a medium for integrating the various trust-related data, exploiting the network effect of information availability and accessibility, combining it in ways that can lead to better trust assessments, hence improved security and dependability for a range of transactions conducted via the Internet. The goal here is not necessarily to use RDF in place of the various security specifications, but to use it as a "meeting place" where information from different specifications can be combined. This seems to be an important facility for assessing and building a reasonable basis for trust. The consistent use of URIs ⊣ [5] to identify important concepts goes a long way to facilitating such use.

Looking forward, we aim to develop ontologies, RDF vocabularies and rules for making trust assessments based on existing security standards and other available information, and use these to demonstrate ways to improve security and dependability in open networked systems.

---

20. Acknowledgements -

⊣ TOC&nbs p;

Some small sections the text used to describe the various protocols and data formats have been copied verbatim or nearly verbatim from the corresponding specifications cited.

This document has been authored in XML using the format described in RFC 2629 ⊣ [9], and converted to HTML using the XML2RFC utility developed by Marshall Rose (⊣ http://xml.resource.org/).

References -

⊣ TOC&nbs p;

[1] ⊣ Kohl, J. and ⊣ B. Neuman, "⊣ The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[2] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Mess age Bodies", RFC 2045, November 1996.

[3] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2 046, November 1996.

[4] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[5] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resou rce Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[6] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[7] Klyne, G., "A Syntax for Describing Media Feature Sets", RFC 2533, March 1999.

[8] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[9] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[10] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

[11] Klyne, G., "Protocol-independent Content Negotiation Framework", RFC 2703, September 1999.

[12] Elkins, M., Del Torto, D., Levien, R. and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[13] Eastlake, D., Reagle, J. and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", RFC 3275, March 2002.

[14] Housley, R., Polk, W., Ford, W. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[15] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (2nd ed)", W3C Recommendation xml, October 2000.

[16] Thompson, H., Beech, D., Maloney, M. and N. Mendelsohn, "XML Schema Part 1: Structures", W3C REC-xmlschema-1, May 2001.

[17] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes", W3C REC-xmlschema-2, May 2001.

[18] Bray, T., Hollander, D. and A. Layman, "Namespaces in XML", W3C Recommendation xml-names, January 1999.

[19] Boyer, J., "Canonical XML Version 1.0", W3C Recommendation xpath, March 2001.

[20] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", W3C Recommendation xpath, November 1999.

[21] Clark, J., "XSL Transformations (XSLT) Version 1.0", W3C Recommendation xslt, November 1999.

[22] Lassila, O. and R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification" , W3C Recommendation rdf-syntax, February 1999.

[23] Brickley, D. and R. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation, February 10, 2004.

[24] Eastlake, D., Reagle , J. and D. Solo, "XML-Signature Syntax and P rocessing", W3C Recommendation xmldsig-core, October 2000.

[25] Eastlake, D. and J. Reagle , "XML Encryption Syntax and Processing", W3C Candidate Recommendation xmlen c-core, August 2002

[26] ⊣ Ford, W., ⊣ Hallam-Baker, P., ⊣ Fox, B., ⊣ Dillaway, B., ⊣ LaMacchia, B., ⊣ Epstein, J. and ⊣ J. Lapp, "⊣ XML Key Management Specification (XKMS)", W3C Note xkms , March 2001.

[27] ⊣ Miller, J., ⊣ Resnick, P. and ⊣ D. Singer, "⊣ Platform for In ternet Content Selection (PICS) Rating Services and Rating Systems", W3C Recommendation REC-PICS-services, October 1996.

[28] ⊣ Miller, J., ⊣ Krauskopf, T., ⊣ Resnick, P. and ⊣ W. Treese, "⊣ PICS Label Distribution Label Syntax and Communication Protocols", W3C Recommendation REC-PICS-labels, October 1996.

[29] ⊣ Evans, C., ⊣ Feather, C., ⊣ Hopmann, A., ⊣ Presler-Marshall, M. and P. Resnick, "⊣ PICSRules 1.1", W3C Recommendation REC-PICSRules, December 1997.

[30] ⊣ Chu , Y., ⊣ DesAutels, P., ⊣ LaMacchia, B. and ⊣ P. Lipp, "⊣ PICS Signed Labels (DSig) 1.0 Specification", W3C Recommendation REC-DSig-label, May 1998.

[31] ⊣ Marchiori, M., Cranor, L., Langheinrich, M., ⊣ Presler-Marshall, M. and J. Reagle, "⊣ The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", W3C Recommendation REC-PICS-services, April 2002.

[32] ⊣ McBride, B., ⊣ Wenning, R. and L. Cranor, "⊣ An RDF Schema for P3P", W3C Note P3P-rdfschema, January 2002.

[33] ⊣ Klyne, G., ⊣ Reynolds, F., ⊣ Woodrow, C. and ⊣ H. Ohta, "⊣ Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies", W3C Working Draft CCPP-struct-vocab, March 2001.

[34] ⊣ Gudgin, M., ⊣ Hadley, M., ⊣ Mendelsohn, N., ⊣ Moreau, J. and H. Nielsen, "⊣ SOAP Version 1.2 Part 1: Messaging Framework", W3C Working Draft soap12-part1, June 2002.

[35] ⊣ Gudgin, M., ⊣ Hadley, M., ⊣ Mendelsohn, N., ⊣ Moreau, J. and H. Nielsen, "⊣ SOAP Version 1.2 Part 2: Adjuncts", W3C Working Draft soap12-part2, June 2002.

[36] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D., Patel-Schneider, P. and L. Stein, "⊣ DAML+OIL (March 2001) Reference Description", W3C Note daml+oil-reference, Decem ber 2001.

[37] ⊣ Hallam-Baker, P. and ⊣ E. Maler, "⊣ Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)", OASIS Committee Specification sstc-core, May 2002.

[38] ⊣ Godik, S. and ⊣ T. Moses, "⊣ OASIS eXtensible Access Control Markup La nguage (XACML)", OASIS Committee Secification cs-xacml-specification-1.0, November 2002.

[39] Hallam-Baker, P., Kaler, C., Monzillo, R. and A. Nadalin, "⊣ Web Services Security Core Specification", OASIS Working Draft wd-wss-core-04, Novembe r 2002.

[40] International International Telephone and Telegraph Consultative Committee, "Specification of Abstract Syntax Notation One (ASN.1)", CCITT Recommendation X.208, November 1988.

[41] International Telephone and Telegraph Consultative Committee, "Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", CCITT Recommendation X.209, 1988.

[42] International International Telephone and Telegraph Consultative Committee, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT Recommendation X.509, November 1988.

[43] International International Telephone and Telegraph Consultative Committee, "Specification of Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", CCITT Recommendation X.680, July 1994.

[44] International International Telephone and Telegraph Consultative Committee, "Abstract Syntax Notation One (ASN.1): Specification of basic notation", CCITT Recommendation X.680, July 2002.

[45] International International Telephone and Telegraph Consultative Committee, "ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", CCITT Recommendation X.690, July 1994.

[46] International International Telephone and Telegraph Consultative Committee, "ASN.1 encoding rules: Specification of basic encoding Rules (BER), Canonical encoding rules (CER) and Distinguished encoding rules (DER)", CCITT Recommendation X.690, July 2002.

[47] Schneier, B., "Applied Crypography", John Wiley and Sons ISBN 0-471-11709-9, 1996.

[48] Sloman, M., "Policy Driven Management for Distributed Systems", Journal of Network and Systems Management, Plenum Press Volume 2, part 4, 1994.

[49] Wireless Application Group, "UAPROF User Agent Profile Specification", WAG WAP-174, November 1999.

Author's Address -

TOC&nbsp;

Graham Klyne
Nine by Nine
14 Chambrai Close
Appleford
Abingdon, Oxon OX14 4NT
UK
EMail: GK-SWAD-E@ninebynine.org
URI: http://www.ninebynine.net/

Appendix A. Revision history -

TOC&nbsp;

2002-09-24:

- *Document initially created.*

2002-10-02:

- *Added section describing framework for interaction between autonomous communicating components.*
- *Updated references.*
- *Revised abstract and introduction.*
- *Editorial revisions.*
- *Added additional placeholder sections and references.*

2002-10-14:

- *Added sections describing OpenPGP and XML signature.*
- *Added placeholder section for XKMS.*
- *Added several references to W3C supporting specifications.*

2002-10-16:

- *Added sections describing XML encryption and XKMS.*
- *Added placeholder section for Kerberos.*

2002-10-23:

- *Added sections describing Kerberos, SAML and XACML.*
- *Refined some text in the section on XKMS.*
- *Added several references to ITU supporting specifications.*

2002-10-24:

- *Added sections describing PICS, P3P and CC/PP.*
- *Added more references to W3C specifications.*

2002-11-08:

- *Editorial pass over entire document: various improvements* *to text.*

- *Added forward-looking note to conclusions.*
- *Added to-do list of desired improvements.*

2002-12-02:

- *Fixed up diagrams in sections 2.1, 2.2. Uses later version of RFX2XML to include them inline.*
- *Improve description of XACML, based on a better understanding of its underlying copncepts.*
- *Added description of OASIS Web Services Security (WSS) specification.*

2002-12-18:

- *Added CCLRC copyright notice.*

TOC&nbsp;