Chemical Markup, XML and the World-Wide Web. Part III: Towards a Signed Semantic Chemical Web of Trust.

Georgios V. Gkoutos [1], Peter Murray-Rust [2], Henry S. Rzepa [1] and Michael Wright [1]

(1) Department of Chemistry, Imperial College of Science, Technology and Medicine, UK - (2) School of Pharmaceutical Sciences, University of Nottingham, UK - **November 29, 2000**

We describe how a collection of documents expressed in XML-conforming languages such as CML and XHTML can be authenticated against digital signatures which make use established X.509 certificate technology. These can be associated either with specific nodes in the XML document or with the entire document. We illustrate this with two examples. An entire journal article expressed in XML has its individual components signed by separate authors, and the collection is placed in an envelope and again signed. The second example involves using a software robot agent to acquire a collection of documents from a specified URL, to perform various operations and transformations on the content, including expressing molecules in CML, to automatically sign the various components and deposit the result in repository. We argue that these operations can be used to build what we term a authenticated and semantic chemical web of trust.

Show Index Show Signature

Introduction

A prominent characteristic of modern chemistry is the increasing automation of processes such as synthetic combinatorial chemistry, or of data capture by instruments associated with high-throughput screening. These developments presage the era of intelligent robotic chemistry where components of the system are programmed to make decisions from structured information and to issue instructions to other components using structured messages. An example of this might be a robotic chemical synthesiser automatically identifying and scanning a newly published journal article for interesting molecules reported there, and either ordering them for testing or deciding to synthesise them from existing starting materials.[1]

Part of this vision stems from that of the "Semantic Web" proposed by Berners-Lee,[2] where he conceives of simple robots or agents which can accurately gather and act upon high-quality data found on the Web or other networks. Web-based resources ("URLs") would contain "metadata" describing the resource, e.g. the authorship, ownership and access rights of the resource.[3] [4] Metadata can also be used to make assertions about other forms of metadata such as *"resource X conforms to protocol Y"*, or *"I confirm that resource Z was authored by organisation Q"*. Part of the necessary process to define and identify what we mean by the term "high-quality" data is to establish a so-called "web of trust", which would be based on certifiable metadata and information objects. In other words, can we, or the agents that act on our behalf, trust the authorship and content of metadata statements?

Chemistry as a discipline is well suited to this sort of reasoning. Chemical entities are well understood and described by nomenclature that is in the public domain. It is meaningful to make a series of related assertions including, but not limited to, the following;

1. "Agent H identifies material K from an article in a journal authored by F and published by G"
2. "Agent H works/acts for organisation E"
3. "Agent H decides that material K, which corresponds to a molecular connection table X, will be required"
4. "To be useful, K must have < 1% impurity"
5. "Acetylsalicylic acid matches connection table X"

6. "Authority M states that aspirin is a common name for acetylsalicylic acid"
7. "Aspirin is entry A-312 in the catalogue from supplier J"
8. "Supplier J asserts that compound A-312 is > 99% pure"
9. "Y represents the NMR spectrum of compound A-312 in chloroform as solvent"
10. "Computer program Z analyses Y as having < 1% impurity"
11. "Program Z was written by organisation Q"
12. "Q is on E's list of approved NMR software suppliers"
13. "J is on E's list of approved compound suppliers"
14. "M is on E's list of approved chemical metadata suppliers"
15. "G is on E's list of approved chemical publishers"
16. "F is on H's list of known/trusted authors"

From this, a semantic chain can be constructed which automatically leads to Agent H proposing an order of compound A-312 from supplier J. Whilst in most chemical organisations Agent H is currently likely to be a research scientist, delegation of such purchasing decisions to a robot agent is already happening in other sectors of business-to-business (B2B) electronic commerce. We argue that such a "chemical web of trust" is a realistic vision, but is mainly dependent on the ability of suppliers to use a uniform web-based approach to data and metadata. In this context, we recently proposed [5] one such universal approach to Web-based Chemistry using XML (eXtensible Markup Language), and a specific implementation of XML termed CML (Chemical Markup Language).[6] Here we describe a mechanism for the certification of chemical data and metadata which could be used to implement this vision.

Authentication of Chemical Information

We require the following components to authenticate information:

a. **A list of authorities whom we are prepared to accept as making assertions we can act upon.** These assertions may be about "facts" or may authenticate assertions made by authorities that we do not as yet accept. For example if we assert *"The absolute conformation of (-)-labetalol is (R,R)"* the reader should not accept this without confirmation. However much greater confidence is generated by accepting it from the assertions:
   1. In *Acta Cryst* (1984), **C40**, 825, the authors state that the absolute configuration of (-)-labetalol hydrochloride is (R,R).
   2. The International Union of Crystallography requires that the reporting of absolute configurations by X-ray crystallography in *Acta Cryst* conforms to its guidelines.
   3. The editorial office of *Acta Cryst* use algorithms to check the data consistency of every published structure.
   4. The editorial office of Acta Cryst have the power to reject publications of absolute configuration which do not meet its guidelines

   The reader may now feel able to accept the assertion with a high degree of confidence (e.g. > 95%).
b. **The ability for each authority to certify their assertions.** The assertions made above cannot be automatically validated without referring back to (printed) material, which itself provides the authentication. When assertions are made in electronic form it is essential that their **origin** is verifiable and that their **content** is shown to be uncorrupted.

Thus we see authors and publishers making authenticable assertions about the validity of their facts and assertions. It is important to distinguish between these concepts:

- **Authenticity** is the ability to verify that a document/assertion has been created by the authority to whom it is attributed and that it is uncorrupted after its creation.

- **Validity** is the ability to show that a validation process has been correctly carried out.

In general, "facts" (primary data) are not validatable other than verifying their authenticity from an authority. However secondary data derived from primary data by a process (often algorithmic), can be validated. Thus *"I mounted crystal X on an X-ray diffractometer and collected diffraction data Y"* is not normally validatable electronically. However *"From diffraction data Y, I used program Z to determine the crystal structure of A as B"* is validatable. It is possible to support or refute this claim by running the same or equivalent computation.

We indicate below some of the electronic components that will need authentication and/or validation:

a. **Chemical structures**, with both two and three dimensional co-ordinates. Validation includes the following hierarchy:
    1. **Syntactic conformance**. We have described earlier how Markup Languages can be used to ensure syntactic validity (i.e. that the components of a chemical structure obey a given CML DTD or Schema). [6] [7] Schemas have greater power than DTDs and we shall describe later how they enhance validation of CML document instances.
    2. **Self-consistency**. CML-based representations of molecules contain some required internal self-consistency (e.g. that the `atomRefs` in bonds must reference valid atoms).
    3. **Regularisation, normalisation and canonicalisation**. We assume algorithms to determine "chemical consistency" such as valence checking, aromaticity and tautomerisation. There is no single approach to this and a variety of protocols will exist.

    These all involve agorithmic "checking of validity" through certified procedures. For example, XML-based validation requires an authentic schema and authenticated output from a validating XML parser. We assume that the appropriate DTD or Schema is authenticated before use and that the output includes this authentication. If the output itself is authenticated, it represents a trustable object for use in the semantic chemical web.

b. **Validation of computations.** Much chemical information now originates in the output of algorithmic-based modelling procedures, which may themselves include access to databases. Such programs will need to be authenticatable and their output certified.

c. **Instrumentation.** As with modelling programs, the parameters defining instrumental procedures need to be declared and the output authenticated.

Markup of Chemical Information

For information to be easily authenticable and certifiable, it must be structured into precisely defined information components or objects, and these definitions must be openly declared and accessible. One such approach has been developed, called **XML** (eXtensible Markup Language). Documents marked-up according to XML specifications have several characteristic features. They comprise data contained within tags known as **elements**, which may themselves be hierarchical. For example in the CML (Chemical Markup Language) specification, [6] a `<molecule>` element may contain many `<atom>` sub-elements or children. Elements can have associated attributes and attribute values and such documents are said to be 'well-formed' if the syntax for expressing the elements, their attributes and their values all follow generic XML specifications.

A precise specification can be created of the allowed elements, their characteristics and the boundaries of the attribute values. This is referred to as a **Document Type Description** (DTD), and it can itself be expressed as an XML-conforming document called a **schema**. Any XML document conforming precisely to a given schema is said to be "valid". Because the structure of the document is precisely specified, this allows transformations of the data contained within it, using eXtensible Style Sheets (XSL) or other tools. An illustration of this process as applied to chemistry is the ChiMeraL Project,[7] where CML based documents are used to express molecules and their properties, and where the data can be visualised using conventional Web browsers and appropriate XSL transformations.

As noted above, XML-based validation will require an authentic schema. For example, CML version 1.0 is defined by a published DTD[6] and only validation against this can certify that documents or components of documents contain authentic CML 1.0

We propose here that authenticity can be ensured by a procedure known as XML signing,[8] which is based on the generation and use of key-pairs contained in so-called X.509 certificates. A description of X.509 certificates and their use for establishing the authenticity of Java applets has been previously described by us in some detail.[10] In the next section, we describe how X.509 certificates can be used to create authenticated XML documents or document components, and how such procedures can in turn be used to create certified documents and components contained within them.

Signed Documents and Document Components

Recent legislation in the UK and elsewhere has granted digital documents similar legal status to printed ones. It is self evident that someone receiving such documents must be able to establish that they are uncorrupted or unaltered since they were created by their originator(s), and that their source is itself authentic and verifiable. The first document format to become accepted in this context is Adobe Acrobat PDF (Portable Document Format). For example, virtually all electronic journals now offer articles in this format, and other important applications of PDF include patent submissions and compound safety sheets. An Acrobat PDF file can be digitally signed using X.509 certificates which can serve to ensure that they, like the Java applets noted above, are authentic. However, an important limitation of the Acrobat format is that the digital signature can only be applied to the entire PDF document. Authenticating smaller components of such a document is not possible, which would exclude a typical requirement for e.g. multiple authors to individually sign their component.

We propose that these and other limitations of Acrobat-format files can be overcome by using XML documents containing XML signatures. As examples we will use two sources of information. The first is an article describing the ChiMeraL project[7] and available in XML form as supplemental information associated with the journal. This article contains not merely components authored by different people, but has well defined data components originating from different sources, and which we therefore use as a model for thechemical publication of the future. The trust associated with such an article originates at least in part from the original authors. In this example this is individually signed by the authors, and hence is not an automated procedure. The second example derives from the use of an Web-based agent or robot which has been programmed to automatically traverse a collection of documents from a remote site and on the basis of pre-defined procedures and associated algorithms, to express the documnets in valid XML form and to create specific metadata for the collection. The procedures include automatically validating any specific occurrences of e.g. an MDL Molfile and converting these files to valid CML. These components are then automatically signed by the robot agent as certifying that the metadata and converted CML files were produced by this procedure and no other. The authors of the robot and the procedures it implements have in effect granted the robot a proxy to use their certificates.

Internal XML Signature

We note firstly that the procedures described here make use of experimental tools and standards which have not yet completed full ratification, and so should be regarded as being only illustrative of the concepts involved rather than as a definitive formula for its implementation.

The procedure starts by generating key pairs for three agents (in this case, each author of the article). Each key-pair is password protected (and known only to each author) and saved within a keystore. Each pair consists of a private key (which remains locked in the keystore) and a public key, both being required to sign a document. The keys are used to produce an X.509 certificate, which might be uniquely associated with either an individual or an organisation. By embedding this certificate within an object known as a signature, the receiver is able to verify that the signature and any object or objects the signature contains is unaltered since the time it was signed. To achieve the greatest degree of trust, X.509 certificates, like other legal document such as e.g. driving licenses or passports, should carry a stamp of approval from a trusted agency. The certificate generated using the process above is known as a 'self-signed' type, and as such would rely upon any receiver trusting (or being able to verify) the signing agent is who or what they claim to be. To achieve a higher degree of trust, the X.509 certificate can be submitted to a so-called certification agency (CA) along with proof of identity, which may indeed be a passport or driving license. The CA will add its own certificate to that of the applicant (known as a root certificate) and return the compound certificate. A CA can operate globally (such as Verisign or GlobalSign) or it may be purely organisational. Such certificates therefore allow a "chain of trust" to be established, a chain which any recipient of a certificate can follow themselves if they wish to do so.

The ChiMeraL XML document contains several overall components, each of which can individually validated against a different XML Schema. We have chosen to illustrate the signing process by selecting four components of the original document.

1. a molecule expressed using standard CML,
2. a spectrum using an extended CML schema,
3. a reaction expressed using an extended CML schema,
4. the document abstract expressed as XHTML.

Each component can in principle be signed by a different agent. For example, the spectrum could be produced directly from an instrument capable of expressing its output as XML and carrying an internal X.509 certificate, whilst the molecule co-ordinates might be generated from a database search or a molecular modelling calculation, again each containing implicit X.509 certificates. In our case, since no such instrumental, database or program certificate infra-structure exists yet, we have emulated the process by installing our own "self-generated" certificates. We note that by retaining these signatures throughout its lifetime, the document could contain an audit trail for each of its components.

In the next stage, the generated keys are used to sign the individual document components and the resulting signatures are stored as XML nodes using a proposed XML signature language.[10] These signatures can either contain, or be contained by, the components they are signing. In either case, each signature must contain links to the component it is presumed to be authenticating (normally expressed as a URI), as illustrated in Figure 1.

Figure 1: *This enveloping signature contains the component it is signing (shown in bold) within a `dsig:Object` element.*

```
<dsig:Signature>
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/WD-xml-c14n-20001011"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <dsig:Reference URI="#sign_molecule">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>kYI/aeHODcfdAVwfd5D+iifwamo=</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>
    DNRqWAjLILQnoPxiWvXfhZlTApoH2CxEfPW4BxLGuUm4oU7loJ9Tdg==
  </dsig:SignatureValue>
  <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <DSAKeyValue>
      <P>
        /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9s
        ubVWzXgTuAHTRv8mZgt2u...
  </KeyInfo>
  <dsig:Object Id="sign_molecule">
    <molecule title="tetrahydrofuran">...</molecule>
  </dsig:Object>
</dsig:Signature>
```

Two forms are possible, an enveloping signature (Figure 1) and an enveloped signature (Figure 2). For the former, the reference URI refers to an object with an `dsig:Id` attribute that matches it, and the element to be signed is contained by the enveloping signature element. This produces a "human-readable" document where the signed component is readily identifiable. In contrast, the enveloped signature uses a URI to define the component being signed, and hence can reside anywhere in the document, for example at the end. Although less readable, it is easier to automate the insertion of such signatures. An example is the signed version of the CML schema (http://www.xml-cml.org/cml_schema_ie_02_signed.xml).

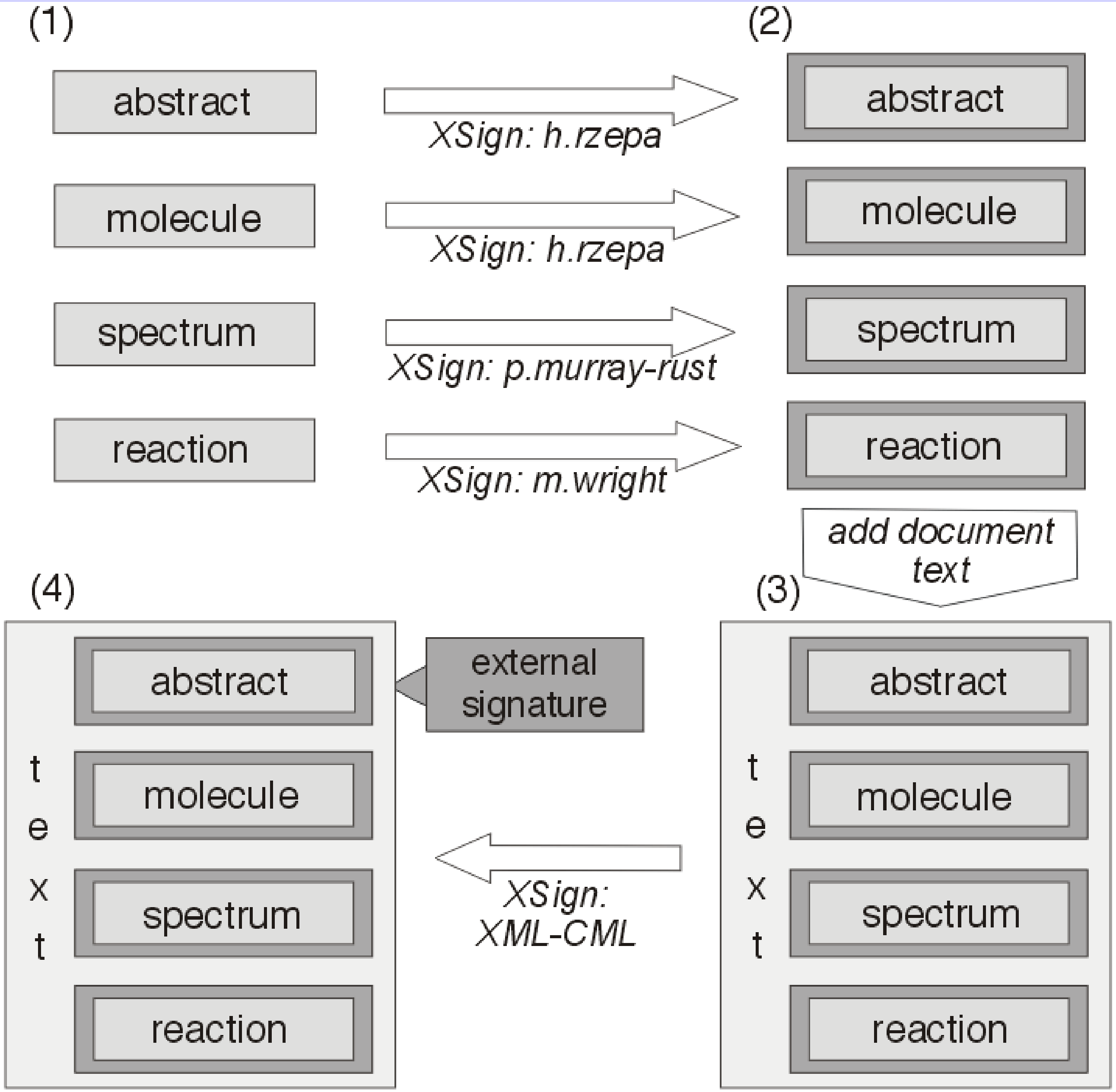
Figure 2: *This enveloped signature has an empty URI attribute and hence signs the rest of the document (shown in bold).*

```
<molecule title="tetrahydrofuran">
  <formula>C4 H8 O</formula>
  <string title="CAS">109-99-9</string>
  ...
  <dsig:Signature>
    <dsig:SignedInfo>
      <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/WD-xml-c14n-20001011"/>
      <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
      <dsig:Reference URI="">
        <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <dsig:DigestValue>qqEOUky3GP1WCzwjlp0xAwzsHUg=</dsig:DigestValue>
      </dsig:Reference>
    </dsig:SignedInfo>
    <dsig:SignatureValue>
      iIFvJvG9epMKZ1zTPvmBgy5XE5hyq+vL6kfp9SSuWEaxchy61Ychfg==
    </dsig:SignatureValue>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <DSAKeyValue>
      <P>
        /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9s
        ubVWzXgTuAHTRv8mZgt2u...
  </KeyInfo>
  </dsig:Signature>
</molecule>
```

The procedure shown for creating such signatures is illustrated schematically in Figure 3, and the results can be seen as part of the supplemental information. The signing procedure authenticates two different aspects of the document:

a. Firstly, the structure of the document itself is authenticated. This includes the element signed, and any sub-elements it contains, along with all attributes of these elements and their values. This procedure ignores the order of any element attributes, and any "white space" or line-breaks in the specifications of the elements and their attributes, which is normalised using the built-in XML parser of the signing/authenticating tool. White space is however considered significant in the values of each attribute.
b. Next, the content itself of each element is signed. The specific tools that we used to do this include significant white space and also the platform-specific line breaks. In general, line breaks often carry semantics, particularly when "pre-formatting" is used in the document and so must be retained. A more problematic issue is whether white space is normalised to a single character, or left un-normalised. Typically, an XML-aware editor might normalise white space, whereas the signing tool will assume it is un-normalised.

Figure 3: *Signing procedure for the ChiMeraL document: (1) Four document components are prepared as separate XML files. (2) Each component is individually signed, using the authors' certificates and enveloping signatures. (3) The components are concatenated and combined with the remaining text. (4) The completed document is over-signed using an external signature. (SVG version of this figure)*



A recipient receiving a signed XML document has several possible actions. A simple tree-view display of the XML in a browser such as Internet Explorer 5 will reveal the location and number of signed components. If the XML document uses (for example) an XSL stylesheet transformation of the document to achive an on-screen display, then current versions of browsers need not in general show the presence of any signatures. One might expect that future versions of browsers would automatically detect signatures in XML documents and allow the reader an option to view the associated X.509 certificate and their verification if required. Current generations of browsers do indeed support this feature if present in Java applets and email. An interim solution might be to separately process the document using the authentication tools provided as part of the IBM signing suite, prior to viewing the document within a browser. It is also possible to pre-process the document using a stylesheet such that any certificate present is displayed in the browser window. An illustration of this particular process can be viewed in the supplemental information associated with this article.

External XML Signatures

Internal signatures are best suited for documents that have been completed and are unlikely to change. Adding further signatures may alter the content of pre-existing signatures. If a document is under development by multiple authors, then internal signatures are less appropriate, since each would have to be removed, recalculated and replaced each time its component is changed. An alternative approach is to use external signatures, which makes use of separate XML files linked to their signed components by a URI. One can envisage a database holding XML documents along with their collection of external signatures. Retaining superseded signatures and documents automatically supplies an audit trail. Because authentication using this procedure must be initiated from the signature document, there is currently no implicit way to tell if a document or document component has been externally signed.

Figure 4: *A detached signature is contained within a seperate XML file. A URI attribute (shown in bold) links this file to the document or document component it is signing.*

```
<dsig:Signature>
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/WD-xml-c14n-20001011"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <dsig:Reference URI="http://www.ch.ic.ac.uk/chimeral/xsign/article.xml">
        <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <dsig:DigestValue>QLfOmhf0CVTJeGRi0RGlqrJ7phQ=</dsig:DigestValue>
      </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>
    IzYvVLmiUtXeJzFZJMFni1w1CqePQyrALpvxhZD/Y2R4D7XR3Xcchg==
  </dsig:SignatureValue>
  <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <DSAKeyValue>
        <P>
          /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9s
          ubVWzXgTuAHTRv8mZgt2u...
    </KeyInfo>
</dsig:Signature>
```

Document Encryption

The act of signing a document only serves to authenticate it. If a greater degree of protection is required, the document or its components can also be encrypted. This can be done using two mechanisms. The simplest, and oldest technique is using a password devised by the encrypter, and which itself needs be sent to any person wishing to view the document. This password must therefore be deemed potentially insecure, since it might be intercepted on its way to whoever needs to read the document. An example of an XML component encrypted in this manner is seen in Figure 5.

Figure 5: *Encrypted XML element*

```
<docml:document>
  <EncryptedElement algorithm="DES/CBC/PKCS5Padding"
        contentType="text/xml" encoding="base64" iv="/w8IHCxetvQ=">
    PFXFr/Cr7UTcJTQXEg3KiCqRyLkHzi3IF+KtJJY7eTs61lUgw54LKokxW4WLOwSrUwLZQ0sf0K5/
    St9mmu7cHxNhP2WLQ0xBj6QZ8kONMawGNeAqpR/EUpcjsxxhDjc5JoU9IhVGIVJHJr0/98AWtkS8
    4oTCl9M0CBRBFMzbqG6HB+afVUYCaZTDb9XYqUduOP/sbOiSnJarNPg0Z9Lv8XbxVZEp24CmuoOE
    6vpPYOAyKAcFiNY7oUQtmv4QoEwB+YsxBOrpAHfgAdokK6unr415atyzxP0OAt90topPRFBHXLAA
    ...
  </EncryptedElement>
</docml:document>
```

A more secure method of encryption would be to make use of an X.509 certificate sent to the encrypter by the intended recipient of the document. This certificate contains a public key which is used to encrypt the document instead of a password. The safety of the document is ensured because only the original owner of the certificate has the corresponding private key, which is the only mechanism which can be used to decrypt the document fragment. Because this process does not involve password transmission it is far more secure, but by its very nature it is less well suited if many recipients would need to read the document.

Access Controls

Both methods described previously would be cumbersome in a large organisation where multiple documents and their components would need to be read by many people. We note, but do not illustrate, an interesting intermediate solution which can be implemented for XML documents. XACL (XML Access Control Language) aims at providing XML documents with a sophisticated access control model and access control specification language. With this technology, access control policies control how an XML document appears to the end reader. This system defines who is able to read, change, add or delete nodes in an XML document and can be set to log all requests and changes. It is based around the following XML files

- Target. This is the 'source' document, containing appropriately marked up information. This should be stored in a server-based database and should be accessible only through the XACL protocol.
- Policy. This contains a set of access rules declaring what privileges each user or group of users has over each node on the target document. It should be stored with the target document.
- Request. This is a document sent to the XACL engine by a user wishing to access the target document. It declares who the user is, which node they wish to access and what they wish to do. On receiving the request, the XACL engine will look up the access rules in the policy file and return the following.
- Decision_list. This declares whether or not each request has been passed. These decisions are also logged
- View. This is the subset of the target document that the user has been allowed to access, and is dynamically generated from the document database and sent to the requester's browser. The rest of the document remains hidden.

Agent-based Signing Procedures

The signing procedures above were applied manually to a single document and its components by the authors of that document. An alternative approach is to delegate the signing process to an agent. This agent can perform a task pre-programmed by its author/creator. In our case, we illustrate this by using the **ChemDig**, **JChemTidy** and **JChemMeta** agents described previously.[11] [12] This involves using a tree traversing robot (ChemDig) to retrieve the HTML content of a remote site and to perform three operations on each document located in the tree:

1. Each HTML document is normalised to conform to XHTML syntax (JChemTidy), producing a well formed and valid XML document.
2. Any chemical data files transcluded into the original HTML document are then parsed, and key chemical meta-information (JChemMeta) is either extracted from the original file or derived using suitable software (e.g. a canonical SMILES string is produced using a specified algorithm). This meta-information can then be expressed into the XHTML document via an appropriately defined schema.
3. Finally, selected types of transcluded chemical files such as the MDL Molfile are converted to a CML representation according to published Molfile specifications using a conversion tool written by us.

The next stage involves establishing a level of trust that the conversions and transformations of the chemical data indeed involve authentic processes as described by us. To achieve this, we have now added two further steps in the sequence indicated above:

1. The derived meta-information as contained in the two elements `<meta>...</meta>` and `<link>...</link>` and the converted XHTML is signed as described in Figure 2.
2. Each of the XML/CML files converted from a Molfile is similarly signed, resulting in a collection of XHTML and XML documents each of which can be authenticated against a formal digital signature. This authentication signifies that a specific Molfile to CML conversion process has been conducted, and that meta-information has been added according to our specifications. These components, if added to a larger database of chemical information would automatically carry their provenance and authenticity.

To illustrate the complete process, we have taken another recently published article[13] . The supplemental information for this article contains a collection of HTML documents, together with 13 MDL Molfiles which have been linked (transcluded) to the main document. We have now charged our ChemDig agent/robot to produce from this a well formed and valid XHTML document from each original HTML file, to add specific chemical metadata to the XHTML, and to transform the 13 original Molfiles to valid CML form. This process creates in effect a set of assertions about the operations conducted on the original information. It does of course, not validate the original chemical claims made. The resulting collection of documents is deposited as supplemental information with the present article. .

Conclusion

The techniques described in this article will allow an entire document expressed in XML, or individual components of such a document, or both, to be authenticated against a unique digital signature encapsulated in what is referred to as an X.509 certificate. This represents a considerable advance over the use of Acrobat PDF files for producing authenticatable documents. The various ways of implementing signatures allow for a diverse set of applications of this technology. Most importantly, the signature can be used to authenticate the specific schema and hence XML language used to encapsulate the information in the document, in the form of a signature attached to the document schema itself. In this manner, the recipient can be assured that the very structure of the document is authentic, as well as the data it contains.

Chemical information and data could be assembled from a variety of sources, including instruments, computational procedures, databases, e-journals and e-books, and the authenticity of each source individually signed. Because each signature is individually a well formed and valid component of the overall XML document, it can be retained as an audit trail of the source of the information it refers to. As such, the signature might authenticate that component, or it may indicate that the information has been deliberately or otherwise either altered or superseded. Signatures can be also regarded as envelopes. Thus small components of a CML document which contain one or more molecules and associated properties can be wrapped in a large envelope containing in effect a database of such molecules. At the other extreme, individual properties such as e.g a melting point could be signed if deemed necessary.

The diversity of information sources may require a context or role to be placed on the signature. Thus an individual might sign a component containing molecular co-ordinates in a different role to e.g. authorising the abstract of the document. Role signatures are very new and we may expect rapid progress in this area.

The signatures can be included within the document itself if the nature of the document is such that it is not intended to be further edited. This would represent an archive of the information. Such documents are extensively used in areas such as patent submissions and drug submissions. If the document is intended to be extensively revised, then the signatures can be deposited in an external database, with URI links to the components they refer to. This approach might be useful for establishing a clear audit trail of the various information components in an organisation, or of the process representing the eventual publication of a journal article. The signing procedures could be extended to two other areas, namely access control lists for individual document components, and encryption of these components to ensure that only the intended recipient can make use of them.

We suggest that the concept of information component authentication and delivery to intended recipients represents a radical departure from current methods of document transmission and processing. We anticipate the techniques describe here will have widespread importance in the industrial chemical and publishing industries, and could ultimately serve as an infra-strucutre on which to base the creation of a chemical and semantic web of trust associated with on-line information.

Acknowledgement

One of us (GVG) thanks Merck Sharpe and Dohm and the EPSRC for the award of a studentship.

References

This article expressed as XML, along with a number of examples are available via the supplemental information pages at http://www.ch.ic.ac.uk/chimeral/documents/xsign/

1. A similar process was described by H. S. Rzepa, in "The Internet: A Guide for Chemists", Ed. S. Bachrach, *American Chemical Society,* 1995, **Chapter 11,**
2. T. Berners-Lee, M. Fischetti and M. Dertouzos, *Harper, San Francisco*, 1999, *(www)*
3. Resource Description Framework (W3C), *(http://www.w3.org/RDF/)*
4. Dublin Core Metadata Initiative, *(http://purl.org/DC/)*
5. P. Murray-Rust, H. S. Rzepa, M. Wright and S. Zara, *Chem. Commun.*, 2000, 1471-1472
6. For a formal description of CML version 1.0, see P. Murray-Rust and H. S. Rzepa, *J. Chem. Inf. Comp. Sci.*, 1999, **39,** 928 *(www)*
7. P. Murray-Rust, H. S. Rzepa and M. Wright, *N. J. Chem.*, 2000, **in press,**
8. W3C Working draft - "XML-Signature Syntax and Processing", *(http://www.w3.org/TR/xmldsig-core/)*
9. ChiMeraL - CML Development, *(http://www.ch.ic.ac.uk/chimeral/)*
10. A. P. Tonge, H. S. Rzepa and Hiroshi Yoshida, *J. Chem. Inf. Comp. Sci.*, 1999, **39,** 483-490
11. G. Gkoutos, P. Kenway and H. S. Rzepa, *J. Chem. Inf. Comp. Sci.*, 2000, **in press,**
12. G. Gkoutos, P. Kenway and H. S. Rzepa, *N. J. Chem.*, 2000, **submitted,**
13. S. Martin-Santamaria and H. S. Rzepa, *J. Chem. Soc. Per. Trans. 2,* 2000, 2378