The Implementation of a System for Evaluating Trust in a PKI Environment

E.Ball, D.W Chadwick, A. Basden, Information Systems Institute, University of Salford, Salford M5 4WT

1.Abstract

This paper describes a system that allows the trust index of a Certification Authority (CA) to be computed both statically and dynamically. Static calculation is based on a CA's published Certificate Policy (CP) and Certification Practice Statement (CPS), whilst dynamic calculation is based on the actual current practices of the CA. At the heart of the system is an expert system that has knowledge about the factors that are important in computing the trust in a CA. Static calculation may be performed in one of two ways. In Method 1, the expert system asks the user (the CA's relying party) a series of questions, which he can answer by consulting the published CP/CPS of the CA. In Method 2, the expert system asks the same questions to a CPS Server, which takes its answers from an XML formatted CPS. This requires the CA administrator to first produce an XML formatted CPS, which we describe, and publish this in its LDAP directory along with its public key certificates and revocation lists. We describe the CPS server, which retrieves the XML CPS's as signed attribute certificates, and feeds answers to the questions posed by the expert system using a Simple SOAP protocol that we have designed. Dynamic calculation of the trust index may be based on information gathered from up to five sources: an Audit Certificate created by the external auditors of the CA, dynamic performance monitoring of the CA's rate of publication of Certificate Revocation Lists, information gathered by the relying party, information gathered by the subscriber, and information gathered about the vendor of the CA's PKI software. We have currently implemented the first two of these. The software has been written in Java and also provides tools that enable Audit Certificates and CPSs to be prepared and published.

2.Introduction

In a Public Key Infrastructure (PKI), users hold public-private key pairs. A PKI user might be an application, a person (or more precisely an application acting on behalf of a person), or an electronic device, in fact any entity capable of communicating electronically. Each PKI user has an electronic name used to identify it in communications. This name might be for example the IP address of a device, the LDAP distinguished name of a person, the email address of an application, or a pseudonym of someone wishing to remain anonymous. A Certification Authority (CA) is the entity responsible for binding the public keys of the users to their electronic names. This binding is digitally signed by the CA to prove its authenticity, and the construct is called a public key certificate, or certificate for short. The international standard format for public key certificates is defined in X.509 [1]. A user may send a message digitally signed with their private key to a receiver (the relying party), and the relying party (RP) can validate the user's digital signature providing they have a copy of the user's public key or certificate. Public key certificates may be easily and securely distributed and published on the Internet, since the signature of the CA renders them unforgeable and unmodifiable without detection, providing the RP has a trusted copy of the public key of the CA. Many Web browsers and email clients come pre-configured with dozens of CA public keys. However, the RP has no way of knowing how trustworthy each of these CAs is.

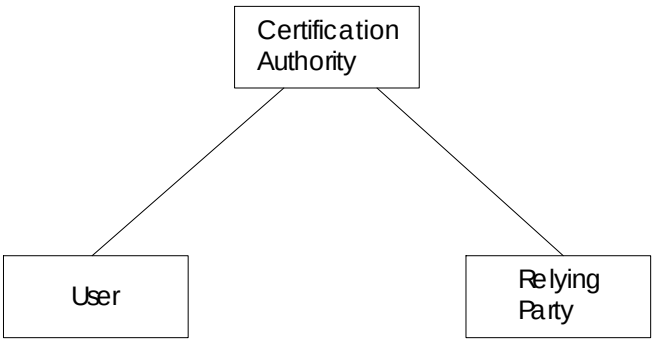

Figure 1. The participating entities

The degree to which a RP can trust the binding of the public key to the electronic name of the user, and that the electronic name is the correct one for that user (i.e. by this we mean that the electronic name for end entity A is not purposefully bound to the public key of end entity B by an untrustworthy CA, thereby allowing B to impersonate A) depends upon many factors that are ultimately related to the trustworthiness of the CA. Clearly if the user and the RP are the subjects of the same CA, then the task is relatively easy, as the RP has knowledge about its own CA and its trustworthiness. But as the number of PKIs grows, and as e-commerce and inter-organisational data transfers increase, one can increasingly expect the user and the RP to be the subjects of different CAs. Thus there will be a need for a RP to depend upon, or trust, a foreign CA. Just how is a RP supposed to base its trust decisions on a remote CA that might be completely unknown to it? Chokhani and Ford [2] have enumerated a long list of factors that CAs need to take into account when producing a trustworthy and secure PKI environment. These factors include the practices followed by the CA in authenticating the users; the CA's operating policy, procedures, and security controls; the user's obligations (for example, in protecting their private key from use by anyone other than themselves); and the stated undertakings and legal obligations of the CA (for example, warranties and limitations on liability). Our research has tried to quantify those factors that go towards building the trust that might be placed in a remote CA by a relying party.

Note that the PKI is only concerned with establishing the identity of users of the CA i.e. authentication. There are many other factors involving the level of trust between the user and the relying party, for example the authorisation rights, and these will depend on the nature of the relationship between them. For example in an e-commerce relationship a provider of goods will be interested not only in the correct identification of a user, but also in his credit worthiness and his track record e.g. the number of purchases that the user has previously made without refuting any of them. Such considerations are beyond the scope of the present paper but have been addressed in Manchala [3].

The system described here was produced as part of the Intelligent Computation of Trust project. In this project the trustworthiness of a CA was evaluated by an expert system, IStar [4, 19], developed at the University of Salford.

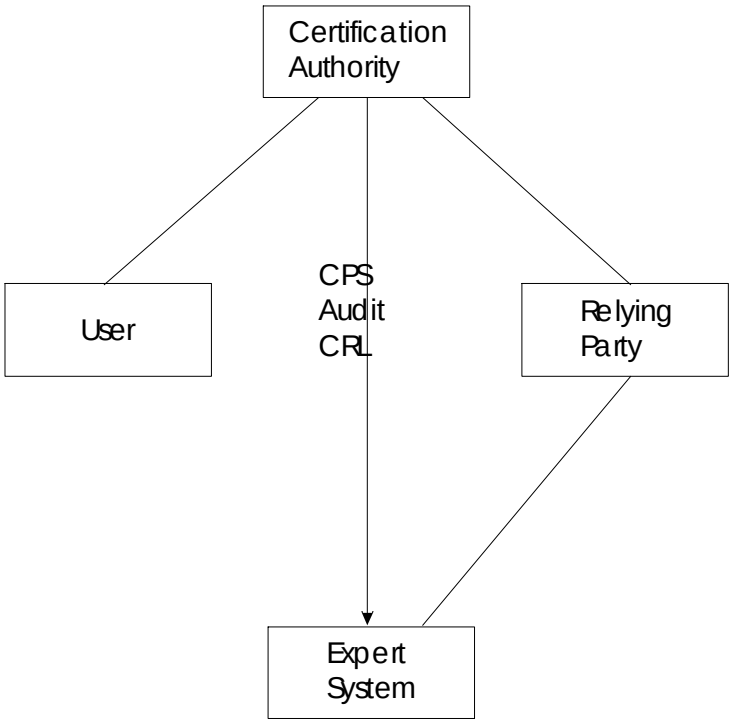

Figure 2. Calculating Trust

The system operates in two separate stages. The first stage, called the Static Trust Calculation stage, evaluates trust based on the information that the CA publishes about the way that it operates. This information is obtained from the CA's Certificate Policy (CP), which states *what* the certificate can be

used for, and the CA's Certification Practice Statement (CPS), which describes *how* the CA operates in order to fulfil its policy, see figure 2.

However in this case we are relying on the CA to be truthful in its statements, and to reliably follow its own stated policies and procedures. Clearly a less than honourable CA will not be completely accurate in its statements, and therefore the Static Trust Calculation will not accurately reflect the trustworthiness of the CA. Consequently we need external independent verification about what the CA actually does in practice. The second stage, called Dynamic Trust Checking, evaluates trust based on the actual performance of the CA. This is obtained from three separate sources: what the RP already knows about the CA, what the CA's external auditor publishes about the CA's operations, and finally what the CA makes known about itself through publication of its Certification Revocation Lists (CRLs) (see figure 2).

3.Static Trust Calculation

3.1.By Reference to a Textual CPS

Chokani and Ford have given a framework for the contents of Certification Practice Statements and Certificate Policies in [2]. This framework is a text based method for describing the contents of a CPS and CP. CA administrators will typically use the framework to construct their own CPSs and CPs. Initially, for the Static Trust Calculation, the expert system asks the RP a number of questions, and these are answered by reference to the published CPS and/or CP of the CA. The questions have been based on Chokhani and Ford's framework. Examples of typical questions asked by the expert system and their allowed range of answers are given in table 1.

Question	**Answer**
Who generates the subject's signing key pair	User or CA
To what extent do you feel that the CPS handles the authentication of persons adequately	0, 1, (low) 2, 3, 4, 5, 6, 7, 8, (high) 9, 10
To what extent are damages covered in the CPS?	0, 1, (low) 2, 3, 4, 5, 6, 7, 8, (high) 9, 10
Are loss limitations mentioned	Yes/No

Table 1. Example Expert System Questions

The questions are asked as the expert system starts its inference processing. Within the expert system is an inference net, in which the expert knowledge is held within the nodes of the net and the way that they are interconnected together by arcs. Functional blocks model Bayesian, Boolean, Probabilistic , Arithmetic and other types of variables. The blocks are interconnected via weighted links and this weighted network is used to infer the trust index. Nodes represent factors important to the calculation of trust. Arcs represent the interplay of the factors and the consequences preceding factors have on the following ones. This depends upon the values of the factors and the way they are combined together using Bayesian, Boolean, Probabilistic or other functions. During the inference process, the value for a factor is determined from the answer given by the RP to the node question, as shown in table 1. The weightings of the various factors were determined by PKI experts prior to inference, and are hard coded into the system. There were many issued raised in the design and construction of the inference net, for example, how should multiple input arcs into one node be combined in order to produce the output arc, and how many output arcs should there be from a particular node. The way that we designed the inference net, and how we addressed the many issues that were raised, are described in [5]. Once the inference net was built, we then had to determine the weightings of the various factors, one against another. This was done by interviewing PKI experts and asking them to rank the various factors against each other in order of importance, on a scale from 1 to 10. The experts could agree that some factors were more important in the calculation of trust than others, but for other factors there was no general agreement. A description of our interviews and findings, and the way that we analysed these results, is described in [6]. The consensus view of the experts, or our preferred ranking when there was no consensus, was then fed into the inference net, so that it was able to calculate a value of a CA's trustworthiness, based upon the answers provided from the CP and CPS by the RP.

Textual CPSs and CPs can be read and understood by humans, but they are quite difficult to understand, and it is certainly time consuming to consult them in order to answer the complete set of questions asked by the expert system. We therefore do not think that many RPs will be prepared to

invest their time in reading the CPSs in order to answer the questions posed by the expert system, as shown in Method 1 of figure 3.

3.2.By Reference to a Structured CPS

Since textual CPs and CPSs are very difficult, if not impossible, to interpret automatically using computer software, we need to be able to construct CPSs in a structured format that can be easily read and interpreted by computer software. In this way, the CA can produce its CPS in structured format just once, can publish it on the Internet, then countless RPs can download it, feed it into our expert system via a CPS server, and be given an automatic static calculation of the trustworthiness of the CA (see Method 2, figure 3).

Extensible Mark-up Language (XML) is becoming a popular and standard way of producing machine-readable documents, and so we decided to use XML to represent a structured CPS. XML is defined by the W3C in [7]. XML is a mark-up language that is able to give a tree shaped structure to a document. Nodes of this tree comprise sections of text of the document. Nodes are also able to have attributes that describe some feature or property of the node, for example, a *document title* node might have a font attribute with value Arial 16. Being a tree structure nodes are also able to contain other nodes.

An example of a simple leaf node in XML is:
<xmltag attribute1=”somevalue”, attribute2=”somevalue” >
The text body of the node.
</xmltag>

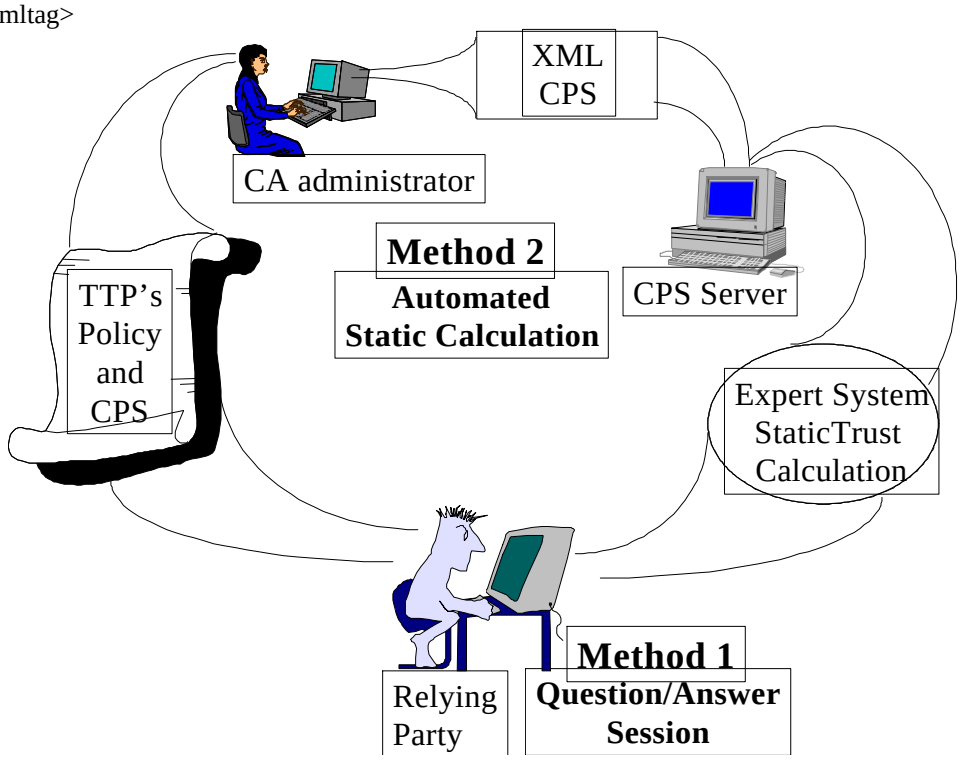

Figure 3. Static methods of trust calculation

We have mapped the Chokani and Ford CPS framework into this structure by assigning XML tags to the section headings of the framework, by using the text body to store some reference text that explains the node to a human reader, and by assigning attributes to the nodes to represent the values of the nodes. If an attribute is attached to a non-leaf node, then this represents a property of the subtree contained within this node. It is unusual to have attributes assigned to non-leaf nodes, since this negates the purpose of having attributes attached to the subordinate leaf nodes.

For example if we pick “section 2, General”, of Chokhani and Ford, this node would have an XML tag of <General>, but would not have any associated text or attributes as it is simply a top level section heading i.e. it is a container node. Within this section are others e.g. “2.2 Liability”. This node would have a corresponding XML tag of <Liability>, and within that “2.2.1 CALiability”, would have an XML tag of <CALiability>. Within this section would be defined all the leaf nodes corresponding to

the things that make up the CALiability. The basic template in Chokani and Ford does not go further into the detail of this section, but we have extended the sub-section headings where necessary e.g. 2.2.1.1 CAWarranties, is a sub-section that defines whether or not the CA has warranties. An example of how this is converted into an XML leaf node is as follows:

```
<CAWarranties Answer= "yes">Does the CA have
warranties?</CAWarranties>
```

Note that we have used an attribute called Answer to actually define the contents of this section of the CPS. In this example, the XML CPS states that the CA does have warranties. If however, a different CA did not offer any warranties, then the Answer in its XML CPS would be “no”. The text body is an explanation of the meaning of the node, needed in case the XML document is rendered for human viewing or checking. This is typically, but not necessarily, in the form of a question. An example XML CPS can be download from [20].

3.2.2.XML checking and Signatures

The structure of an XML document may be formally defined using a number of techniques, such as a Document Type Definition (DTD) [8] or XML Schema [9]. At the time of carrying out our work we decided that a DTD would be used, as the XML Schema was still in the process of final definition as a standard. Consequently a DTD definition of the XML CPS was produced by our Java tool at the same time as the XML CPS was produced (see later). To give an example of the DTD, the CA Warranties node in the DTD would be constructed as:

```
<!ELEMENT CAWarranties (#PCDATA)>
<!ATTLIST CAWarranties Answer (yes|no) “no” >
```

This restricts the corresponding XML CPS node to having just one attribute, called Answer, whose allowed values are “yes” or “no”, the latter being the default. Having the DTD also allows the XML CPS to be used by XML tools (editors and viewers) such as Xeena [10]. An example CPS DTD can be download from [20].

In addition to the DTD we have also given the CA administrator the ability to add the CA’s digital signature to the XML CPS that is produced, so that the authenticity of the CPS can be verified subsequently by any RP. This stops an attacker from being able to edit an XML CPS distributed on the Internet without detection. Without the signature, and attacker could, for example, edit an XML CPS to make the CA appear to be less (or more) trustworthy than it actually is. Although a standard has now been proposed for the signing of XML [11], at the time of the research this was not finished and supporting tools did not exist. For these reasons we choose not to use XML signatures, but rather digitally signed attribute certificates (see below), since these were already standardised as part of X.509 [1] and we had software tools to support them.

3.2.3.CPS publication

Once a signed XML CPS has been produced, this can be published anywhere on the Internet without fear of it being altered without detection. The CPS could be published as a web page, retrievable via a standard URI. Alternatively, it could be stored as an attribute, retrievable from an LDAP directory. As we are working with public key infrastructures, in which it is customary to store public key certificates in LDAP directories, we chose to store the CPSs and DTDs as embedded within X.509 attributeCertificateAttributes. The attribute certificate construct has the advantage that it is an open standard syntax, and contains within it the digital signature of the contents.

We have defined two new LDAP attribute types, the xML-CPS attribute that holds the CPS in XML format, and the cPS-DTD attribute that holds the DTD. The formal definitions of these attributes [18] are:

(1.2.826.0.1.3344810.1.1.4 NAME 'xML-CPS' EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

(1.2.826.0.1.3344810.1.1.5 NAME 'cPS-DTD' EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

These attributes are then embedded within a single signed attributeCertificateAttribute.

3.2.4.XML DTD Generation

We have built a tool that allows XML CPSs and DTDs to be created. The tool models the XML nodes as Java serialised objects. We have used a hierarchical DNS like representation to address each node in the XML tree. Thus the CA Warranties section used in the above examples would be addressed as:

General.Liability.CALiability.CAWarranties

The user creates a list of Java objects. Each object contains the hierarchical name of the node, the text body for the CPS, and the allowable attributes and their values for the DTD. The relationship between the objects is effectively defined by their hierarchical names. The objects are saved to disc. By clicking the “create DTD” button, the user causes the tool to read in the objects and using their hierarchical names, it builds a tree corresponding to the XML document. The DTD is then generated by walking this tree and outputting the appropriate syntax.

3.2.5.XML CPS Generation

Any XML editor, e.g. Xeena can be used by the CA administrator to construct their XML CPS from our DTD. (Of course, the clever CA administrator could always simply use a text editor and write their CPS in XML, but if they did this there would be no automatic checks on its correctness.) The editor typically displays the DTD elements, the CA administrator chooses those most appropriate for his CPS, and for each element the editor displays the attributes with their allowed answers. The CA administrator simply chooses the answer nearest to that contained in their textual CPS. The editor then writes out the appropriate XML node. The CA administrator can optionally add the text body to each node prior to it being written out. In order to save the CA administrator from this tedious step, we have written a bespoke XML CPS editor, based on our Java serialised objects, that already has the text bodies contained within them.

3.3.The CPS Server

In order for the expert system to operate it needs to ask a series of questions, the answers to which are provided by either the RP via the user interface (Method 1) or via a protocol interaction with a CPS server (Method 2). In the latter case a protocol, based on XML, has been devised so that the expert system can retrieve parts of the XML CPS. When the expert system needs information from the CPS server it first establishes a connection with the server using a prearranged fixed TCP port number (this could be registered with the ICANN). The server replies with a new dynamically allocated port number from a pool available to it. This is to be used by the expert system for the remaining dialogue, and allows the CPS server to be multi-tasking. The expert system then establishes a new session with the CPS server at this new port number. The first data to be transferred over the new session identifies the CA and its LDAP repository, in the form of a URL [13]. This enables the CPS server to retrieve the CPS attributeCertificateAttribute from the LDAP directory. Within the session the expert system then asks XML formatted messages, using the dotted name notation described above to specify the question being asked. The CPS Server can then locate the correct node in the XML tree, retrieve the Answer attribute, and reply to each question. An example of a dialogue is shown in figure 4.

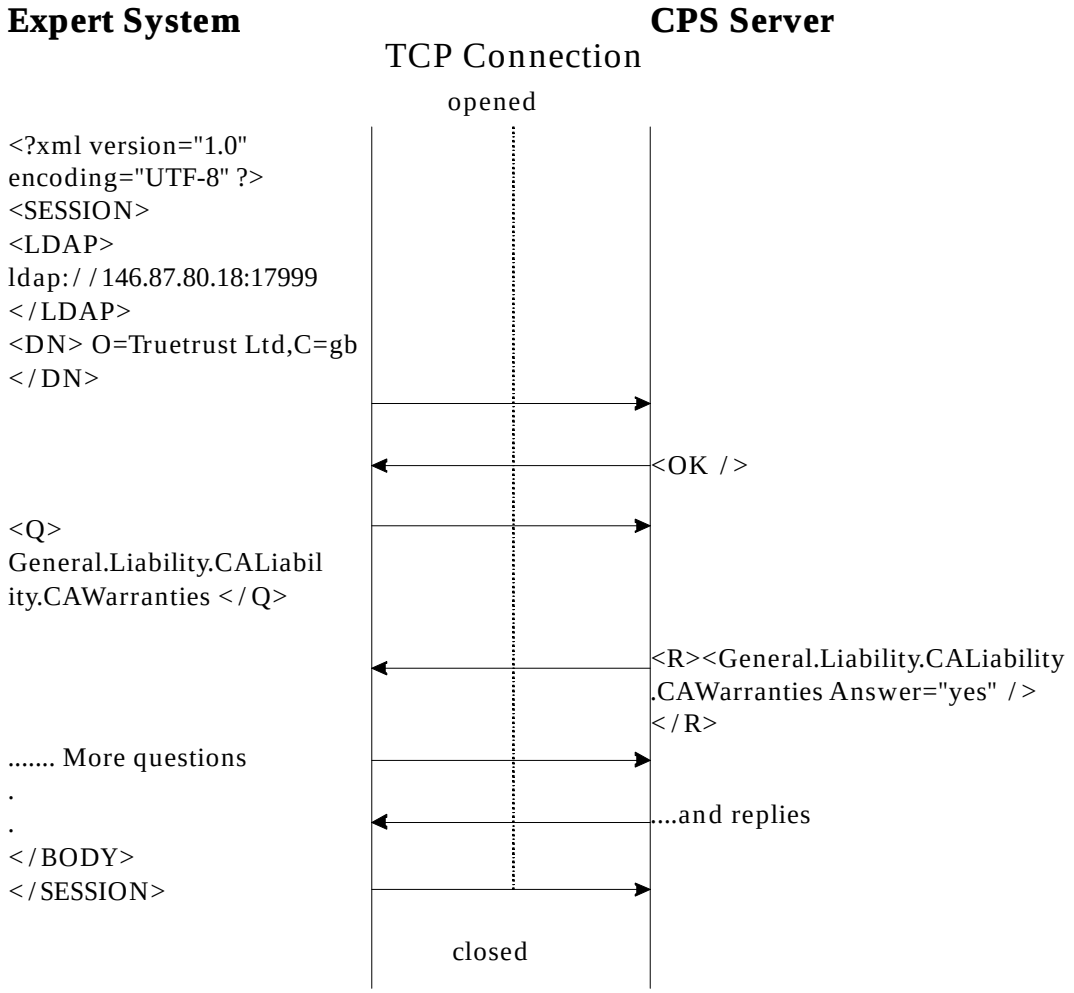

Figure 4. A protocol exchange between ISTAR and the CPS Server.

The above protocol is, in effect, a much-simplified version of the SOAP protocol [14], which was being developed at the same time as our research. We have named our protocol Simple SOAP. SOAP is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. Although the SOAP functionality is much larger than the present requirements, it would be relatively easy to migrate to the use of SOAP at a later date.

The components of the CPS server are shown in Figure 5. It comprises the CPS Manager that is responsible for fetching the XML CPSs, the Session Manager that is responsible for managing the multi-tasking environment, the Parser that parses the Simple SOAP messages, and the Session Database that holds ready prepared answers based on the current XML CPS.

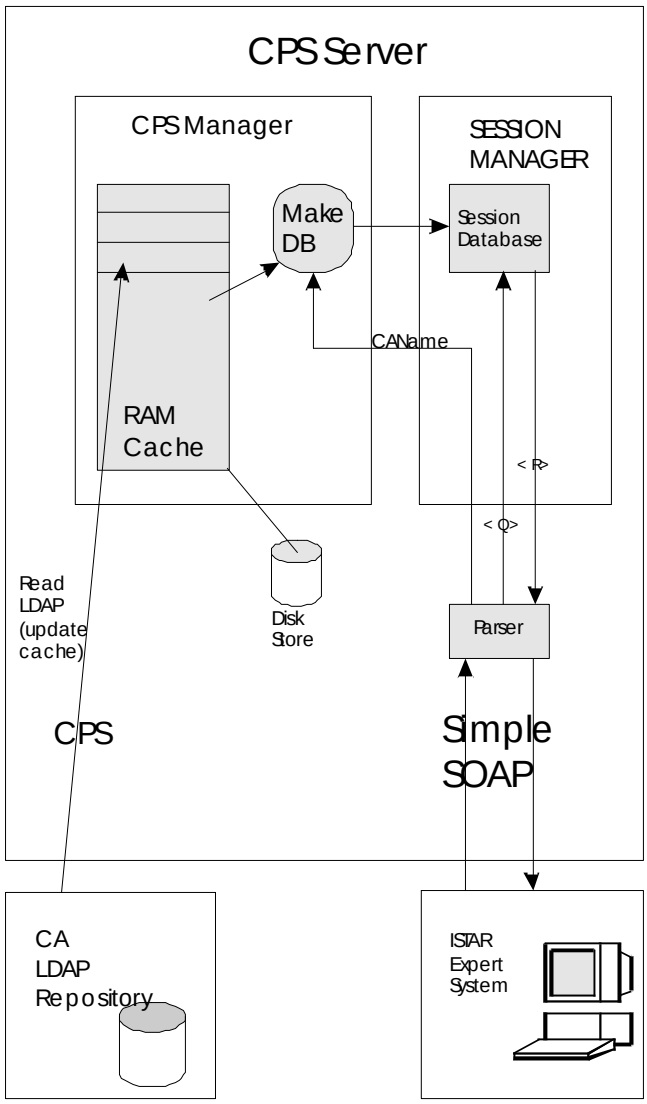

Figure 5. The components of the CPS Server

3.3.2.CPS Manager

The CPS Manager is responsible for obtaining data from XML formatted CPSs. Once a session with Istar has been started, the URI of the LDAP repository, and the DN of the CA are passed to it from Istar. The CPS manager has two ways of getting the CPS data for a CA: either it already has it stored in its local cache, or if not, it retrieves it from the CA's LDAP repository. The CPS Manager first checks its local cache. There are three possibilities:

a) The CPS for the CA is not in the cache at all,
b) It is there but its time-to-live has expired
c) It is there and still within its time-to-live period.

For a) and b) the CA data in the cache must be refreshed. We have used the JNDI toolkit [17] to provide the CPS Manager with an LDAP capability. An LDAP Search request with scope of base object, for base object equal to the DN of the CA, using a filter of attributeCertificateAttribute present, to retrieve the attributeCertificateAttribute, is sent to the LDAP repository. This retrieves the attributeCertificateAttribute, in which should be enclosed the xMLCPS attribute. One problem that we encountered here is that the attributeCertificateAttribute might be multivalued, but we have no way in LDAP of requesting that only the value holding the xMLCPS attribute be returned. Consequently, if multiple attribute values are returned, we have to parse them all looking for the value of interest. In order to solve this problem in future releases of LDAP, we have been working with the IETF PKIX and LDAPEXT groups to specify new Internet RFCs that will allow i) matching rules to be specified for

public key certificates and attribute certificates [15] and ii) allow single attribute values to be returned from multiple valued attributes [16].

Once the correct attribute certificate has been retrieved, the XML CPS is extracted and stored in the cache. It is allocated a time-to-live value based on the validity period of the attribute certificate. The cache is maintained in permanent storage but a copy is held in RAM for faster access.

3.3.3.Session Database

At the instant that the expert system opens a session with the server the data in the cache is locked whilst a Session Database is prepared. The Session Database contains the replies to all the questions that can be asked by the expert system. The replies are taken from the Answer attributes of the XML CPS. The cache is temporary locked to prevent any process writing to it in the middle of the read. Once the Session Database has been created the lock is removed and answers can be sent to Istar.

4. Dynamic Trust Checking

Independent quality assurance, or audit, is needed in order to ensure that a CA faithfully abides by the statements in its CPS. If a CA consistently does what it says it does, then it is as trustworthy as expected from the Static Trust Calculation. Conversely, if a CA fails to carry out the procedures documented in its CPS, it is less trustworthy than expected from the Static Trust Calculation. We have analysed each section of the CPS framework to determine who might perform the quality assurance/audit function of that section, and have determined seven possible QA categories. These are:

- **Assumed Compliance.** No checking needs to be done, as a failure of this factor will not make the CA less trustworthy. An example of this would be the section on Disclaimers and Exclusions. If the CA did not enforce its exclusions of liability this would not lessen one's trust in it (in fact perversely, it might increase one's trust). Actions that have to be performed by the subscriber are also categorised as assumed compliance, since it is the trustworthiness of the CA that we are calculating, and not the trustworthiness of the subscriber. Thus if a subscriber purposefully lets others use his private key, and is willing to assume the consequences of this, the trustworthiness of the CA is not effected by this.
- **Anyone**. Information is in the public domain, and therefore anyone may perform the QA function of this item. An example of this is checking the frequency of publication of the CRLs. We have built a TrustCheck server that performs this function (see later).
- **Subscriber**. A subscriber of the CA may perform the QA of this item. An example of this will be the authentication procedures used by the CA.
- **Relying Party**. The RP may perform the QA function. An example of this is that meaningful unique names are used which allow RPs to easily identify the subject from the contents of the certificate.
- **External Auditor**. The external auditor can perform the checks during one of his audit visits to the CA. Most of the CPS items fall into this category.
- **CA Vendor Audit.** The trustworthiness of the CA depends upon the software being provided by the PKI vendor e.g. the strength of the cryptography, and the randomness of the random number generator. These aspects were judged to be out of scope of the current project, although we could envisage PKI vendors publishing details about their ITSEC, Common Criteria and FIPS 140-1 certifications.
- **No-one.** It is not possible to QA this item. Fortunately we did not find any sections that fell into this category.

Note that some CPS sections may be quality assured by several of the above; it is not a mutually exclusive list.

Most of the CPS quality assurance is performed by an external auditor, who will periodically visit the CA installation and audit its paperwork, computer logs, audit trails, and such like. Most CAs will have a clause about external audit in their CPSs, but at present there is no agreed standard for the level of audit that should take place, nor for the type of audit report that should be produced, nor for where or how the report should be published. Such professionally agreed standards are evolving, e.g. WebTrust in the US [21], and the t-Scheme in the UK [22], and they may eventually be the subject of legislation, as is the case today for example with public limited companies who have to provide annual financial reports. In order to address the current shortcoming, we have made an initial attempt to suggest how auditing might evolve, by defining an electronic Audit Certificate that may be published in an LDAP directory. We have built a TrustCheck server that is capable of downloading information from the

Internet, such as CRLs and Audit Certificates, and can then use this information to respond to the expert system when it asks it to perform a dynamic trust check. The expert system is able to interleave questions to the TrustCheck server and to the relying party, and more over, does so asynchronously, so as to not keep the human RP waiting.

4.1.Audit Certificate

An audit certificate is a variation of an XML CPS. It contains only those CPS items that can be checked by an external auditor. It takes the basic form of the CPS but the questions that are answered by the auditor record what the CA actually did, rather than the CPS statement that records what should be carried out. An example of a node in an audit certificate is:

<AuditLogBackup Answer= “yes”> Have the audit backup procedures been thorough?
</AuditLogBackup>

An example Audit Certificate and its DTD can be download from [20].

We have defined an Audit Certificate LDAP attribute as follows:

(1.2.826.0.1.3344810.1.1.1 NAME 'auditCertificate' EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

and an Audit DTD LDAP attribute:

(1.2.826.0.1.3344810.1.1.3 NAME 'auditDTD' EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

Both of these attributes should be encapsulated in the same attributeCertificateAttribute and digitally signed by the auditor. They can then be held in the auditor’s entry and/or the CA’s entry in an LDAP directory, thereby making them widely available.

4.2.Auditor Software

Using the CPS and DTD Java software described earlier, it is possible to create Java serialised objects for Audit Certificates and their DTDs. Typically we would create the DTD, and then the CA auditor would create the audit certificate based on what he had observed from the CA’s operations. The auditCertificate and auditDTD attributes are then stored in an X509 attribute certificate, the whole is displayed to the auditor and he is then invited to digitally sign it. The attributeCertificateAttribute is then written to the LDAP directory of choice.

4.3.The TrustCheck Server

The TrustCheck server is a more sophisticated version of the CPS server. Instead of downloading and caching XML CPSs when asked to do so, instead it downloads Audit Certificates and CRLs. In addition it continues to poll LDAP repositories, downloading CRLs every time they are published, thereby keeping a statistical record of the reliability of the CA in publishing its CRLs.

The TrustCheck server is also capable of communicating with the expert system using the Simple SOAP protocol that we have devised. This time the expert system asks a subset of the previous questions i.e. only those that can be answered from the Audit Certificate and CRL publications.

A functional breakdown of the TrustCheck server is shown in Figure 6. It comprises a cache that holds information about CAs, a Polling Manager responsible for fetching information from the CAs, a Session Manager responsible for managing the Istar question/answer sessions, and a parser for handling the Simple SOAP messages. For each CA, the cache holds its latest Audit Certificate, and information about their CRL publications. In particular, the cache holds the last n (configurable, but set by us to 50) dates of CRL publication (from the thisUpdate field of the CRL), plus the last CRL retrieved. All of these are combined into a single XML Session Database for the duration of a session between the expert system and the TrustCheck server.

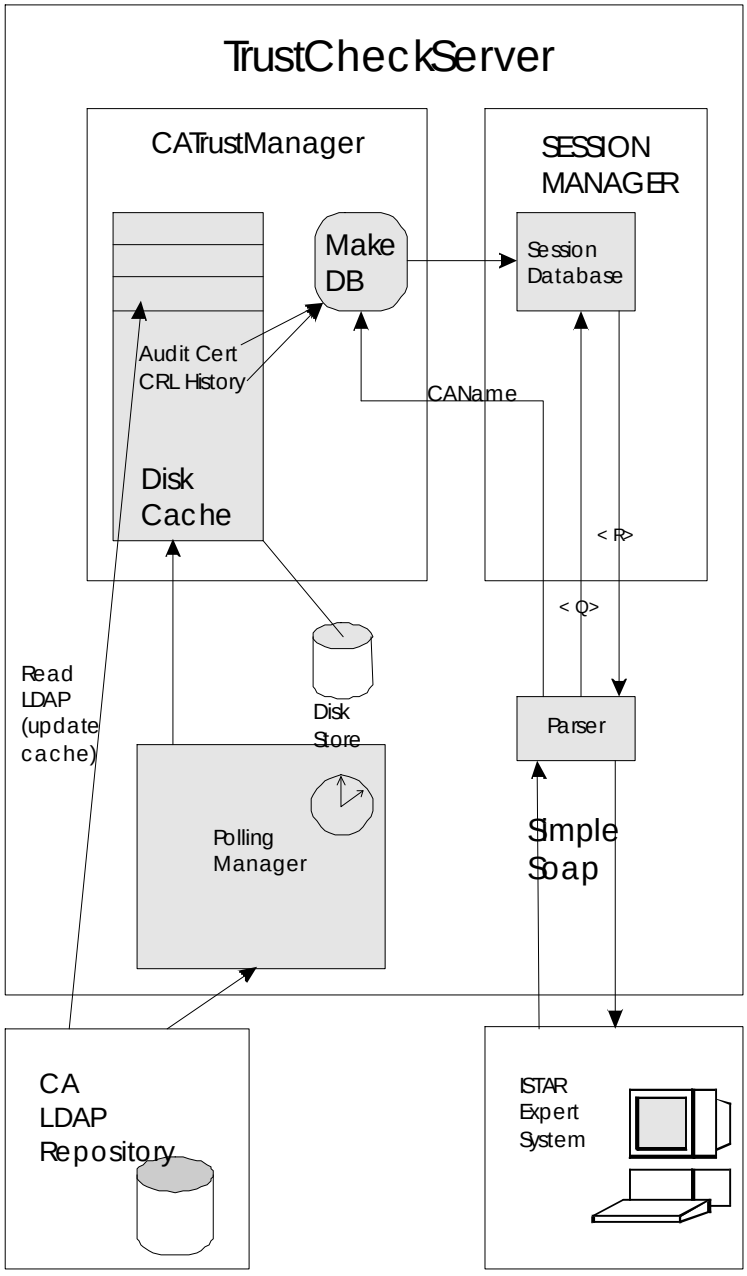

Figure 6. The TrustCheck Server

4.3.2.The Polling Manager

The Polling Manager runs under the control of a timer (currently set to activate every 5 minutes). Each time that it runs it checks the cache and carries out the following actions.
For each CA entry in the cache it checks:
a) The audit expiry date and time (the notAfterTime field)
b) The CRL replacement date and time (the nextUpdate field)

If either time has expired a new attribute certificate or CRL is retrieved. If either cannot be retrieved it will try again in 5 minutes. If the retrieved item is later than the one currently in the cache, the cache is updated. If either the audit certificate has expired or the CRL has not been superceded (i.e. the updated audit certificate or CRL were not published when they should have been), this is a reason to decrease the trust in the CA. The actual decrease is calculated by the expert system. The TrustCheck server simply deletes the expired audit certificate, decreases the frequency of update for the CRLs, plus records that the CRL has not been superseded when it should have been.

In the case of CRLs, the nextUpdate field is optional in X.509. Therefore if it is absent, the Polling Manager estimates the time of publication from the interval between publication of the last two CRLs. (At start up the estimate is set to 24 hours.)

4.4.Non Compliant CAs

In the case where a CA does not have an Audit Certificate because an audit has not yet taken place, and this is confirmed by the CPS, only information regarding CRL publication frequency is available to the session, and the expert system is largely unable to improve upon its static calculation of trust In particular, the cache of the TrustCheck server will only hold the last n (configurable) dates of CRL publication (from the thisUpdate field), plus the last CRL.calculation performed from the CA's published CPS. However if the CPS says that an audit should have been performed, but no Audit Certificate can be found, the expert system will decrease its trust in the CA.

5.Future Work

Currently the expert system and CPS and TrustCheck servers work in close proximity to each other, across a trusted LAN. Thus the Simple SOAP protocol is not protected. Future enhancements would be to add digital signatures and data integrity to the Simple SOAP protocol, and to enhance the expert system so that it is capable of verifying signed messages from the servers. But then we have the tautology of trusting the CAs who issued certificates to these servers in order to compute the trust in other CAs!. Clearly this raises the question of whether the expert system can trust the two servers that are providing it with answers, and indeed, if the RP can trust the expert system that is providing him with answers. Which is just another formulation of the age old question of "who guards the guards?". Ultimately the RP has to trust the reputation of who is providing the quality assurance/audit service in the form of signed attribute certificates, and if the RP can run the servers on his own PC or network then he can be his own root of trust. Future research should explore how reputation can be factored into the system.

Another avenue is to look into the possibility of using the expert system to automatically generate XML versions of CPSs and Audit Certificates. Because the expert system contains very detailed and well tested knowledge of what is important in a CPS and Audit Certificate (from the point of view of trust), once this knowledge has been instantiated by answers from a user (who could be an RP or a CA administrator or the TrustCheck server) it then contains full knowledge of the trust aspects of the CPS and Audit Certificate. Because Istar already has a facility by which knowledge can be exported as ASCII it might be possible to export the knowledge as an XML CPS or Audit Certificate. However there are a number of technical details that remain to be researched, for example, how to turn a directed graph into a hierarchical tree, and what additional knowledge is needed in a CPS in addition to simply calculating the trust in the CA.

When dynamically calculating trust, we have identified five different possible sources of quality assurance/audit information (excluding the no-one and assumed compliance sources). Currently we only use three of these sources, namely: the RP, the auditor and publicly available CRLs. There is scope for increasing this, to include information about the PKI software vendor, and information provided by the CA subject.

We have produced a first version of an Audit Certificate. However this was produced without any input from professional auditing associations. So whilst the concept is novel, the contents will undoubtedly require further iterations from professional computer auditors. Indeed we have already learnt that in financial audits, two reports are produced: a detailed internal confidential one, and a diluted public one. If Audit Certificates were to follow the same route, the public ones may be of significantly less value for dynamically computing trust.

Finally, whilst we have developed a mechanism for computing trust in authentication, we have not yet begun to address the problems of quantifying trust in the level of authorisation needed for e-commerce applications.

6.Conclusions

We have designed a system that can be used as a sound basis for calculating the trustworthiness of CAs in performing their authentication function. We have shown how it is possible to calculate the trust index of a CA both statically and dynamically. At the heart of the system is an expert system (Istar), which is fed by the contents of Certification Practice Statements (in either textual or XML format), CRLs and newly conceived Audit Certificates. We have designed a CPS server and a TrustCheck server that communicate with the expert system via a Simple SOAP protocol. The TrustServer gathers information from Audit Certificates, and also from dynamic performance monitoring of the CA's rate of publication of its CRLs. The software has been written in Java and also provides tools that enables Audit Certificates and XML CPSs to be prepared and published.

References

[1] ISO/ITU-T Rec. X.509(2000) The Directory: Authentication Framework
[2] Chokhani, S., Ford, W. "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework". RFC 2527. March 1999.
[3] Manchala **"**E-Commerce Trust Metrics and Models" IEEE INTERNET COMPUTING March-April 2000 p36-44
[4] Basden A, Brown A J, (1996), Istar - a tool for creative design of knowledge bases, *Expert Systems*, v.13, n.4, pp.259-276, November 1996.
[5] Basden, A., Ball, E., Chadwick, D.W. "Issues Raised in Modelling Trust in a Public Key Infrastructure" Expert Systems, Vol 18, No. 5, (Nov 2001), pp 233-249
[6] Chadwick, D.W. and Basden, A. "Evaluating Trust in a Public Key Certification Authority", Computers and Security, Vol 20, No 7, (Nov 2001) pp 592-611
[7] "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6 October 2000. Download from http://www.w3.org/TR/2000/REC-xml-20001006
[8] DTD http://www.w3.org/TR/2000/REC-xml-20001006
[9] XML Schema http://www.w3.org/XML/Schema
[10] Xeena**,** a visual XML editor from IBM, for editing valid XML documents derived from any valid DTD. See http://www.alphaworks.ibm.com/tech/xeena
[11] D. Eastlake 3rd, J. Reagle, D. Solo. "(Extensible Markup Language) XML-Signature Syntax and Processing". RFC 3275, March 2002.
[12] ISO/ITU-T Rec. X.690, "Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules", 1994.
[13] T. Berners-Lee, L. Masinter, M. McCahill. "Uniform Resource Locators (URL)". RFC 1738, December 1994.
[14] SOAP http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[15] Chadwick, D.W., Legg, S. "Internet X.509 Public Key Infrastructure, Additional LDAP Schema for PKIs and PMIs", <draft-pkix-ldap-schema-01.txt>, September 2000.
[16] Chadwick, D.W., Mullan, S. "Returning Matched Values with LDAPv3", <draft-ietf-ldapext-matchedval-06.txt>, June 2002
[17] JNDI toolkit http://www.javasoft.com/products/jndi/index.html
[18] Wahl, M., Coulbeck, A., Howes, T., Kille, S. "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions". RFC 2252. December 1997.
[19] Basden A, (2000), Some technical and non-technical issues in implementing a knowledge server**,** *Software - Practice and Experience* **30**:1127-1164.
[20] A sample XML CPS, Audit Certificate and their DTDs are available from http://sec.salford.ac.uk/ICT
[21] WebTrust. See http://www.webtrust.org/
[22] The UK t-Scheme. See http://www.tscheme.org/

7.Acknowledgements

This research was carried out under EPSRC grant No GR/L54295, “The Intelligent Computation of Trust”. The authors would also like to thank Entrust Technologies for making available to them their PKI software and toolkits.