On-demand Trust Evaluation

David O'Callaghan and Brian Coghlan

Department of Computer Science
Trinity College Dublin
Dublin 2, Ireland
david.ocallaghan@cs.tcd.ie
coghlan@cs.tcd.ie

***Abstract*— Security is a critical factor in interoperability of grid middleware. There is an opportunity to automate the trust evaluation and accreditation process for grid certification authorities to allow continuous evaluation of their policies and practices. To assess the feasibility and usefulness of automatic evaluation of trust in CAs a trust evaluation system has been designed and a prototype implemented. The service will evaluate a CA based on its published policies and observed practices with respect to a set of rules based on the requirements from an authentication profile. Development and testing are ongoing and we hope to deploy a pilot system shortly.**

I. Introduction

Security is a critical factor in interoperability of grid middleware as higher-level services such as work- and data-management rely on the underlying authentication and authorization systems for their secure operation. An approach to grid interoperability, *MetaGrid*, has been proposed where a number of components provide the central services required for interoperability among grid middleware implementations [1]. Security services will be concentrated, where appropriate, in a central MetaGrid Security Exchange (MSX). In this context, a service for automatic evaluation of trust in grid certification authorities (CAs) is desirable for interoperability at the authentication level as it would allow different grid middleware installations to interact with a common authentication system.

The grid authentication policy management authorities (PMAs) have produced an authentication profile for grid CAs that describes minimum requirements which all CAs must meet in order to be accredited and hence trusted in a grid infrastructure. The PMAs recognise that there is an opportunity to automate the reasonably well-defined trust evaluation and accreditation process for grid CAs. The concept of automatic trust evaluation for grid CAs was introduced in [2]. Here we propose a major extension of that work to allow continuous on-demand evaluation of the CAs' policies and practices. This may then be used by the MSX for evaluation of trust in response to requests from one middleware for credentials to interoperate with another middleware.

To assess the feasibility and usefulness of the above hypotheses a prototype trust evaluation service has been designed and implemented as described in this paper. The software evaluates a CA based on its published policies and observed practices with respect to a set of rules based on the requirements from an authentication profile.

II. Background

A. Grid Certification Authorities and Policy Management

At the start of the European DataGrid project [3] in 2001 it was necessary to create a large international X.509 public key infrastructure (PKI) [4] for grid authentication. The *Certification Authority Coordination Group* (CACG) was established by EDG to coordinate its operation. During the lifetime of EDG, several other international projects adopted its authentication infrastructure and trusted the CAs accredited by CACG.

As EDG drew to a close it became clear that the practices, requirements and policies of the group, and most importantly the established trust relationships, should be carried forward. The *European Policy Management Authority for Grid Authentication in e-Science* (EUGridPMA) [5] was founded, with the blessing of the European Commission's eInfrastructures Reflection Group [6], to coordinate authentication for EGEE[7], DEISA[8], LCG[9] and SEEGRID[10]. The history of the CACG leading up to the formation of EUGridPMA is described in [2]. *The Americas Grid Policy Management Authority* (TAGPMA) [11] and the *Asia-Pacific Grid Policy Management Authority* (APGrid PMA) [12] have since been established to coordinate grid authentication policies in those regions. In October 2005 the *International Grid Trust Federation* (IGTF) was established to coordinate policies and practices between the grid authentication PMAs [13]. The Global Grid Forum [14] has a CA Operations working group to resolve international issues and establish policies and procedures.

B. Organisation of the PKI

In order to discuss trust evaluation it is useful to describe the organisation of the large-scale PKI established by the EDG CACG and coordinated today by the grid authentication PMAs and the IGTF.

The conventional approach to building a large-scale X.509 PKI is to set up a hierarchy of certification authorities with a single root CA and a number of sub-CAs and sub-sub-CAs with various assurance levels, purposes, catchment areas, etc. It is not normally possible to include existing certification authorities into such a hierarchy. When EDG was starting, there were a number of established CAs serving the relevant research and development communities and it was considered

important that these should continue to do so for the project. For this reason, amongst others related to the technical problems of supporting hierarchical PKI with the grid software of the time, a non-hierarchical approach was favoured. There are a number of methods for building a non-hierarchical network of trust between multiple CAs.

In a *cross-signed network of trust*, each CA agrees to sign the root certificates of some or all of the other CAs. An end-entity certificate issued by one CA in the network can be accepted by relying parties of a trusted CA since a path of trust exists from the end-entity certificate, through the cross-signed certificate, to the trusted CA. This requires cross-certification support in the software performing the verification. However, this is not supported by the Globus Security Infrastructure (GSI) [15] and OpenSSL [16] libraries used by the grid middleware unless all the cross-signed certificates issued by the trusted CA are installed as well as the trusted CA's root certificate.

Alternatively a *bridge CA* signs the public keys of CAs that are to be considered equivalent. Each CA also distributes a self-signed root certificate, and a certificate signing the public key of the bridge CA. That is, each CA cross-signs with the bridge CA. An end-entity certificate issued by a CA who is a member of the bridge can be accepted by a relying party who trusts the bridge CA, since the certificate chain provided by the end-entity can be verified all the way back to the issuing CA's root certificate. This is supported in the GSI and OpenSSL libraries only if all the the cross-signed certificates issued by the bridge CA are installed as well as the bridge CA's root certificate and the trusted CA's root certificate. Globus has been tested and found to work in this configuration as described in [17].

Another alternative to a full cross-signed network of trust is a *policy-based network of trust* arrangement, which eliminates the cross-signing aspect. In such a network of trust each CA evaluates every other CA and, if the evaluation is positive, agrees to treat the other CAs as equivalent. Relying parties that want to accept certificates issued by all CAs that are members of the network of trust must install the root certificates for all the CAs. No special software support is required for this kind of network of trust.

In theory, with a bridge or cross-signed network the signature from the bridge CA or a trusted CA is enough to allow the verification of end-entity certificates from all CAs who are cross-signed with the bridge or with each other. In practice however, current software requires that all the cross-signed certificates be installed, as for sub-CAs, and so there is no great advantage in using the more complex bridge and cross-signing approaches. It could be argued that cross-certification poses a greater risk than the policy-based approach since sites in different trust domains (i.e. with a different local CA) install a different collection of cross-signed certificates and this limits the scope to verify the authenticity of these certificates by comparing with copies from other sources.

The $N \times N$ evaluation involved in fully cross-signed or policy-based networks of trust has complexity of $O(N^2)$. However, by establishing a common set of requirements (a threshold) that each CA must meet in order to be accredited in an unsigned network of trust, the complexity can be reduced to $O(N)$. In effect, this puts the policy management authority in the rôle of a *policy-based bridge*. This allows the group to more easily scale up to tens of members and was one of the original reasons for the CACG adopting this alternative and establishing a set of minimum requirements for grid CAs.

C. Requirements and Accreditation

The IGTF authentication profile described in [18] sets the requirements for the policies and operation of 'classic' Certification Authorities. The profile attempts to limit proliferation of grid CAs to one per country, large region or international organisation. CAs are encouraged to establish a wide network of Registration Authorities (RAs). The profile requires that a subject distinguished name is linked to exactly one end-entity for the lifetime of the CA. There is a strong requirement for face-to-face meeting with photographic identification for an RA to verify the identity of an applicant. The CA computer must not be connected to a network unless a suitably secure hardware cryptography module is used. The profile sets requirements for the length of CA and end-entity keys and for the validity period of certificates as well as for certificate extensions. CAs are required to issue CRLs when a revocation is made and on a specified schedule. Other requirements cover the auditability and confidentiality of CAs and their responsibilities with respect to publication and disaster recovery.

The EUGridPMA accreditation process is described in [19]. A new CA wishing to be accredited must distribute draft Certificate Policy and Certification Practice Statement (CP/CPS) documentation [20] to the PMA for comments. The PMA appoints a number of its members to review the CP/CPS in detail. The reviewers provide feedback to the CA on any internal inconsistencies or failure to meet the requirements set out in the authentication profile and the CA has an opportunity to make changes in line with the reviewers' recommendations. Once the documentation is ready, the CA will make a presentation to the PMA members describing in detail "the authentication and vetting procedure and the physical security measures, record persistency, procedures and such." To establish trust, it is important to convince relying parties that each CA was founded in good faith and so the human element is very important. The PMA as a group has the opportunity to question the applicant about the policy and operation of the CA. If the CA is approved then relevant details, including the CA certificate, must be securely conveyed to the PMA Chair for distribution.

The IGTF requirements and accreditation procedures for grid CAs have counterparts in the wider IT context. The *WebTrust Program for Certification Authorities* [21] is one of the more widely recognised accreditation standards for CAs. CA root certificates typically must meet certain standards for inclusion in the trust stores of popular operating systems, web browsers and other applications, such as Microsoft's Windows family and the Mozilla Foundation's suite of products, and

so inclusion can be considered a form of accreditation. The Microsoft Root Certificate Program [22] requires CAs to complete a WebTrust audit or an equivalent third-party audit and also requires that the CA "must provide broad business value to Microsoft platform customers". The Mozilla CA Certificate Policy [23] requires CAs to meet specific WebTrust, ETSI, or ANSI criteria for CA operations.

In general, fully independent third-party audits are not required for grid CAs but there has been some recent work done in GGF by AP Grid PMA members to devise an audit checklist that is based on the WebTrust program and is consistent with the IGTF requirements [24].

Evaluation of trust can be considered a continuous and long-term process — dealing with both changing requirements and changing CA practices — but the current procedures are generally only applied to new CAs looking to be accredited. An automated system would complement the existing manual accreditation process and take some of the effort out of continuous re-evaluation of CAs. The remainder of this paper discusses a proposal for online automated trust evaluation.

III. Principles

A. Static & Dynamic Evaluation

Evaluating CA policy documents can be considered *static* evaluation since it is based on the slowly-changing information published by the CA. Evaluating actual CA practices can be considered *dynamic* evaluation since it is based on information from issued certificates and certificate revocation lists.

Ball, Chadwick & Basden distinguish between *Static Trust Calculation*, based on information published by the CA, and *Dynamic Trust Checking*, where the actual performance of the CA is evaluated [25]. Three sources of information are used for dynamic checking:

> what the [relying party] 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 the publication of its Certification Revocation Lists (CRLs).

Additional sources of information that can be used include the properties of the CA root certificate and end-entity certificates, such as key length, validity period and certificate extensions. These can be compared against the CA's published policy and against the requirements of relying parties. Furthermore, a certificate issued by a CA must have a unique serial number and must fall within the relevant namespace declared by the CA. Certificates can be evaluated on a historical basis to check that all certificates seen from a particular CA meet these requirements. Similarly for CRLs, the update period can be evaluated with reference to the CA's published policy and RP requirements, and also on a historical basis to assess a CA's past performance in this matter.

While some of these factors do not directly relate to the cryptographic 'trustworthiness' of a CA, they *are* considered important for a CA's acceptance by the relying parties.

B. Automatic Evaluation

An automatic evaluation engine was introduced in [2] [1]. As mentioned above each CA must have CP/CPS documentation. Some of the features from these policies and practices were encoded in a CA report file. For the CA report file a basic contextual language involving key-value pairs was used. The language was designed to enable later extension for full expression evaluation, polymorphism and matching capability of a lambda calculus to allow formal analysis, but was initially very simple.

Features are evaluated relative to rulesets. Rules are specific to a particular feature but any number of rules can be defined per feature. The concept of assurance levels is accommodated to allow rulesets to be defined for each level specified by the GGF [26]. Manual third-party evaluations are provided for within an appendix section to each report file.

A *default ruleset* was defined based on the EDG CACG minimum requirements. Each Virtual Organisation (VO) can also define their own rules that override and extend the default ruleset, and each CA can do likewise, overriding and extending the former rulesets. This *Ruleset Inclusion Principle* extends from the general to the specific. It can be extended to users, hosts and even specific services simply by defining the appropriate ruleset. Thus a typical chain obeying this principle might be: default ruleset → VO ruleset → CA ruleset → host ruleset. It is not necessary for a typical subject to have all possible rulesets in their possession, only those rulesets in the inclusion chains that they are interested in.

For an evaluation function f and matrices A, W, R and F of evaluation results, weights, rulesets and features: $A = f(W \times R(F))$. This assumes all possible rules are defined, whereas in practise only the default ruleset is likely to have the full complement of rules defined (which is why the evaluation remains largely an $O(N)$ problem). To cope with this a Boolean matrix D of definition states was added to give: $A = f(W \times R(F), D)$.

IV. Potential Use Cases

A. Trust Matrices

Two forms of graphical trust matrix are also introduced in [2], the *CA feature matrix* and the *CA acceptance matrix*. These aid relying parties in making assessments of CAs. The feature matrix allows an interested party to assess CAs by inspection of the published information and by comparison with other CAs in the matrix. The acceptance matrix allows an interested party to see how CAs evaluate against one or more rulesets.

B. Extending GSI/SSL Authentication

One can conceive of an extended authentication that invokes a trust evaluation (TE) library which evaluates the same information as the matrices and decides whether to trust the authenticating party. The GSI/SSL verification invokes the TE

[1] Online CA Trust Matrices: `http://www.cs.tcd.ie/coghlan/cps-matrix/cps-matrix.cgi`

library with the certificate of the remote entity and the rulesets against which to evaluate it. The TE library finds the CA and extracts other details from the remote certificate and then applies the rulesets and returns a result.

C. Outsourced Trust Evaluation

This may be extended further to a remote online TE service which can be queried during GSI/SSL verification to perform the necessary validation on behalf of the client. Relying parties would communicate with a TE validation service running on behalf of a PMA or VO rather than performing validations with multiple CAs. This could act as a component of the Metagrid Security Exchange mentioned in the introduction.

D. Validated Credentials

One may further extend this to create an online service that issues tokens for validated credentials. A user's client creates a proxy and authenticates with the server, which then performs trust evaluation. The server then signs a certificate extension specifying the rulesets for which evaluation passed. This is returned to the client and added to the proxy. The public key of the validation service must be installed locally in a trusted store on all hosts that trust the service.

A service receiving the credential may check it for a certificate extension signed by a trusted validation service. The certificate extension specifies that the credential passes certain rulesets as evaluated by the trust authority. The choice of rulesets is left to the validation service, which would be appropriate if it is operating at a VO level and sites are willing to accept its authority. If a local override is required the site could run its own (possibly more strict) validation service, or the rulesets could be provided when calling the service.

Alternatively, the credential could specify which rulesets it has been successfully evaluated against and the receiving party in the authentication can decide if it accepts those rulesets. In this case there needs to be a way to uniquely identify rulesets: a hash of the ruleset signed by the validation service would suffice.

This is similar to the CertiVeR/OGRO approach of 'pre-validation' used with OCSP-aware services [27]. This potential use case is not considered further in this paper.

V. Requirements

The software system is divided into a core trust evaluation engine and a number of components which use this engine. Figure 1 gives an overview of the architecture.

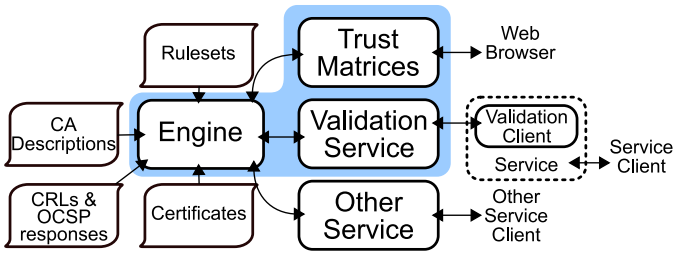

Fig. 1. Trust Evaluation Architecture

A. Engine Design

The original version of the trust matrix software described in Section III-B implemented an acceptance matrix, a feature matrix and a rotated feature matrix. In that design the focus was solely on producing tables for visual inspection. For an on-demand evaluation it is felt that the trust evaluation engine should be designed as a general-purpose library that can be used to produce acceptance matrices and also to provide trust validation services. The CA description language should be hierarchical in structure with key–value pairs at the leaf nodes. There should not be a restrictive pre-defined set of known features which all CA descriptions must contain: the set of features depends on the rules used for evaluation.

Third-party reports on CAs (by auditors, RPs, or other CAs) should be supported as a source of information. There needs to be general support for updating the static information in rulesets and CA descriptions. As well as the static information contained in the CA description the engine should be able to make use of external dynamic information. There must also be logging sufficient for auditing purposes.

Any rule can refer to any number of features and there can be multiple rules defined which refer to the same feature. The result of evaluating one ruleset against one CA should be a list of (⟨rule-name⟩ ⟨eval-result⟩) pairs, e.g. `((Name 1.0) (CP-and-CPS 0.9))` so that the result is identifiable and can be reduced by higher-level functions to give an overall result.

The ruleset is a list of rules or functions that evaluate description features from the CA description. Each rule consists of a function with one or more description features as parameters. The function will typically evaluate the features against some threshold (e.g. that a string is not empty; that a key-size is greater than or equal to 1024 bits).

One desirable feature is that it is possible for the rulesets to be formally analysed to some degree, so a rule function should be self-contained and side-effect free. The ruleset must include an identifier so that it can be specified in an evaluation.

It should be possible to evaluate chained rulesets, e.g. default ruleset → VO ruleset → Site ruleset. There are a variety of approaches to chaining. Rules from a more specific ruleset (e.g. at the site level) could override those from a less specific ruleset (e.g. at the VO level). Alternatively, rules could be composed, passing the result along the chain. Ruleset chains could be transformed to equivalent single rulesets before being applied, or the partial results of an evaluation with one ruleset could be passed to another ruleset on the chain so that results can be added or replaced. Flexibility is paramount.

It is desirable that the engine could be passed a strategy function — by value or by reference (by name) — by its caller. In the case of multi-assurance-level CAs, a set of strategy functions could be passed to be evaluated against. To further complicate matters, each level in a chain might prescribe their own strategy function to apply to subordinate chains.

B. Trust Matrices

It should be possible to display a wide variety of trust matrices useful for visual inspection and assessment of CAs. Feature matrices are required for individual CAs and multiple CAs to allow side-by-side comparison. Acceptance matrices are also required for individual CAs and multiple CAs evaluated against single, multiple or chained rulesets. It should be possible to 'drill down' from a summary of results, via hyperlinks, to individual rule results in a particular CA evaluation.

C. Validation Services

An online validation service should ideally support standard protocols for connections from clients. Suitable protocols include the Online Certificate Status Protocol (OCSP) [28] and the Simple Certificate Validation Protocol (SCVP) [29] from IETF, and the XML Key Management Specification (XKMS) from W3C [30].

D. Formats

As far as possible the file formats for CA descriptions and rulesets should be similar. The files should be human-readable and writable, although in practice special tools may be used. It would seem wise to use or extend an existing syntax such as XML, s-expressions (as in Lisp and Scheme), or Java properties. The format chosen should be extensible. It should be possible to insert new features that have not been previously defined without requiring the software to be rebuilt. In general, the format will consist of a list or tree of keys and values. It would be convenient to use or create tools to validate these files.

CA descriptions or rulesets could be signed to allow for secure third-party use. Alternatively, unsigned CA descriptions and rulesets could be distributed from a trusted source.

It would help if the format for CA descriptions included information from the CA's CP/CPS documents (RFC 2527 or 3647 format) in a canonical form. This might require some mapping between natural language expressions and integer, Boolean or other types.

The CA description should include an identifier so that it can be specified in an evaluation. The Certification Authority's Distinguished Name (DN) is a suitable candidate as this must be unique within a PKI. If more than one CA appears with the same DN then the trust in all of the CAs bearing that DN should be substantially reduced.

E. Programming Interface

The library-level interface needs to be a function to evaluate a specified CA-description against a specified ruleset. For validation with dynamic evaluation the interface should include a certificate parameter.

$$\text{evaluate trust}(\langle \text{ca} \rangle, \langle \text{ruleset} \rangle) \rightarrow \langle \text{return value} \rangle$$

$$\text{evaluate trust}(\langle \text{end entity cert} \rangle, \langle \text{ruleset} \rangle) \rightarrow \langle \text{return value} \rangle$$

The type and meaning of the return value will depend on the ruleset and the algebra.

VI. Implementation

A. Evaluation Engine

The Kawa Scheme compiler [31] was chosen to implement the Trust Evaluation Engine. Kawa compiles Scheme [32] source code to Java Virtual Machine bytecode compatible with the Sun JVM (and others). This makes it attractive for including the Kawa-based engine in Java-based web applications. Kawa can also produce native executables using the GNU Compiler for Java [33]. This is convenient for including the engine in a library to be called from applications using OpenSSL.

A collection of functions was created to handle evaluation of CA descriptions against rulesets. To evaluate a rule, the engine first uses the Scheme `eval` procedure to convert the function body specified in the (non-executable) ruleset to an in-memory procedure at runtime (in this way the engine obtains the ability to compile and run Scheme code at runtime 'for free') and the engine looks up the rule parameter values in the CA description. It then applies the rule function to the parameter values to produce a result. Each rule in a ruleset is evaluated and the results can then be aggregated. Several forms of ruleset chaining are implemented.

Rulesets and CA descriptions are stored in table 'objects' which comprise a local hash-table of data and methods to add entries (directly or from a named file) and to lookup entries given a key.

Some experimentation with memoization of ruleset evaluation has been carried out. In a functional language such as Scheme it is possible to define a generic `memoize` function which can be applied to any referentially-transparent function to produce a memoized version of that function, that is, a version of the function which caches the result for a given set of inputs. This can provide effective optimization without introducing unnecessary complications to the evaluation logic.

An object-oriented wrapper for the trust evaluation engine was created (also in Kawa Scheme) to provide an interface that Java-based programs can call. The engine class has member variables for the ruleset and CA description tables. It also provides methods to perform various types of evaluation.

B. Trust Matrices

The trust matrices web application was developed partly in Kawa Scheme, for the components that interface most closely with the engine, and partly in Java, to be used as Tomcat Java Servlets. A similar approach was taken as for the development of the engine. A collection of functions were built up to produce nicely formated tables from the result of an evaluation by the engine. An object-oriented wrapper was created (also in Kawa Scheme) to provide an nterface for Java-based programs wishing to display trust matrices.

The display tables are constructed from the evaluation results in SXML (XML as S-Expressions) [34]. The SSAX library is then used to conveniently convert the SXML to HTML for display. The HTML includes links from, for example, each ruleset name to an appropriate servlet page showing evaluations using that ruleset.

The servlet loads the available rulesets and CA descriptions into the ruleset and CA description tables of an engine object instance. At present this is a fixed list. The servlet handles requests for various forms of acceptance matrix, that is for evaluations of one or more CAs against one or more rulesets. The URL path specifies the required evaluation:

- **/all/** — evaluate all CAs against all rulesets; a full acceptance matrix
- **/all-summary/** — evaluate all CAs against all rulesets showing a single result value for each CA/ruleset evaluation. This is shown in Figure 2.
- **/ruleset/**<***ruleset name***>**/** — evaluate all CAs against a given ruleset
- **/ca/**<***ca name***>**/** — evaluate a CA against all rulesets
- **/ca-ruleset/**<***ca name***>**/**<***ruleset name***>**/** — evaluate a CA against a given ruleset

The servlet makes calls to the engine to do the requested evaluation and passes the result to the matrix display class to be presented in a suitable HTML form.

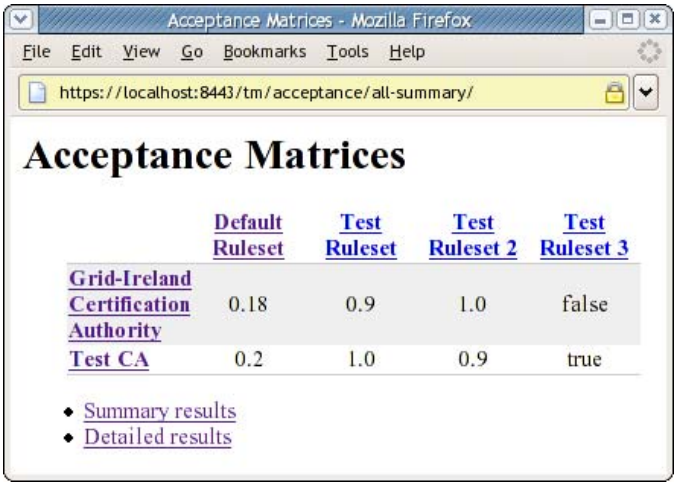

Fig. 2. Acceptance Matrices Screenshot

C. Validation Services

A simple web service has been implemented that accepts a certificate as an argument. It determines which CA issued the certificate and then uses the engine to evaluate trust in the issuing CA. The service returns the result of the evaluation. This demonstrates the use of the trust evaluation engine as a library and is the first step to providing a full online validation service. The existing implementation does not yet support any of the standard protocols mentioned in Section V-C above.

VII. Descriptions and Rulesets

A. CA Description Features

The CA description features correspond to information found in CP and CPS documents and other data published by a CA. The left side of Figure 3 shows part of a CA description for the Grid-Ireland Certification Authority (edited slightly).

B. Rulesets

The rules in a ruleset are based on the requirements a relying party has with respect to CAs. In practice a ruleset would be written by an accreditation body such as EUGridPMA or by a relying party such as a project-specific virtual organisation. The right side of Figure 3 shows part of a ruleset which corresponds to the IGTF authentication profile for classic CAs.

Rulesets and CA descriptions must be semantically compatible with respect to the names, types and values of description features. In the current implementation there is no direct technical support for this: the author of a CA description is expected to take care to make it compatible with the relevant rulesets. In effect, the writer of a ruleset determines what features must be described in a CA description which is evaluated with that ruleset.

C. Ruleset Algebra

The trust evaluation engine does not specify the types of values that can be returned as results of rule evaluations, not the range of values, nor how these values can be combined or reduced to get an overall result for a particular CA with respect to a particular ruleset. The values and the algebra are specified in the ruleset itself.

An important part of constructing a ruleset then is to decide how values are assigned to rule results. To initialise an expert system for trust evaluation Chadwick & Basden [35] took the approach of

> 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. *(Quoted from* [25]*)*

An alternative approach is to set the result values based on values in an appropriate documented standard, if such a thing is available. In the case of a ruleset for the IGTF authentication profile for classic CAs the values have been chosen to approximately match the use of **Must** or **Should** in the authentication profile [36].

The most basic set of rule result values is the Boolean set. Each rule result will indicate if the evaluation of that rule was acceptable or not. Boolean results for an entire ruleset can be reduced to a single result with a logical-AND operator. However, this is a very crude approach as a single *false* value will lead to rejection.

An alternative is to use the range of real numbers 0.0–1.0. This has precedent in probability and fuzzy logic. When working with probabilities, multiplication of independent probabilities will give the combined probability. In fuzzy logic, the fuzzy-AND is usually defined as the minimum value of a set of independent fuzzy set membership functions. An arithmetic mean or weighted arithmetic mean could also be used. Of course, these different functions will produce different overall results and so the assignment of values to rules must be made

```
(ca-description
 (name "Grid-Ireland Certification Authority")
 (dn "/C=IE/O=Grid-Ireland/CN=Grid-Ireland CA")
 (country "Ireland")
 (security-level 'low)
 (CA-email "grid-ireland-ca@cs.tcd.ie")
 (CP-and-CPS
  (RFC-2527-compliant #t)
  (RFC-3647-compliant #f)
  (OID-identifier "1.3.6.1.4.1.10977.10.1.1.0.3")
  (OID-in-cert #f))
 (CA-web-server
  (URL "http://www.cs.tcd.ie/grid-ireland/gi-ca/")
  (cert-publication-max-latency 0) ; < days >
  (CRL-publication-min-freq 23)    ; < days >
  (CRL-publication-max-latency 0)  ; < days >
  (restricted-access #f))
 ...)
```

Example CA Description

```
(ruleset
 (name "Default Ruleset")
 (rules
  (rule
   (name "CP/CPS format")
   (description "CA must have RFC 2527 or 3647 CP/CPS")
   (params ((c2527 (CP-and-CPS RFC-2527-compliant))
            (c3647 (CP-and-CPS RFC-3647-compliant))))
   (func (cond (c3647 1.0)
               (c2527 0.9)
               (else  0.0))) )
  (rule
   (name "CP/CPS OID in cert")
   (description "CA must have OID in cert")
   (params ((oid-in-cert (CP-and-CPS OID-in-cert))))
   (func (if oid-in-cert 1.0 0.2)))
  ...))
```

Example Ruleset

Fig. 3. Example CA Description and Ruleset

with this in mind. Table I shows some examples of combining sets of real numbers with different operators.

TABLE I
COMBINING REAL NUMBERS IN THE RANGE 0.0–1.0

Op.	$\{0.9, 0.9, 0.8, 0.7, 0.9\}$	$\{1.0, 1.0, 1.0, 0.5, 1.0\}$
$\times$	0.37	0.5
min.	0.7	0.5
mean	0.85	0.92

Another approach is to add numerical rule results, possibly followed by a threshold or sigmoid function. If the sum exceeds some threshold then the CA can be accepted. This is similar to the approach taken in spam filters such as SpamAssassin [37], where email messages are assessed against a set of rules, and the results added. Rules check email message for known spam-like features, such as text with the same background and foreground colour, or the same address in both the To and From headers. Un-spam-like features, such as digital signatures, give negative scores. By default, if the message gets a score of 5.0 or more it is flagged as spam.

The trust evaluation system is not limited to simple numerical values. Results can return symbolic values (e.g. `low`, `medium`, `high`); values with attached weights (e.g. `(result (value 0.8) (weight 0.2))`); arbitrary tuples or lists; and unevaluated functions. In each case the aggregation function must be chosen to handle the result types. For some applications, instead of reducing the evaluation result to a single value it may be more appropriate for the result set to be compared against a corresponding acceptance set of individual rule scores for an exact match or partial match. One aspect of our future work will involve evaluating a variety of ruleset algebras.

VIII. Related Work

CertiVeR offers commercial certificate revocation status services using the OCSP protocol and has been promoting the service to the grid community. They have introduced OCSP support for proxy validation, validation policies and 'pre-validation' [27].

Ball, Chadwick & Basden [25] have implemented a trust evaluation system which uses an expert system to evaluate the trust in a CA, based on the CA's CP/CPS (stored in a standard XML format), audit certificates and other sources of information.

The **New Security Infrastructure** project [38] aimed to increase interoperability between PKIs and reduce the complexity of PKI use in applications and services. The main focus of the project was on a *PKI Server* which centralises the tasks of certificate validation, signature verification, certificate retrieval, and certificate path building [39]. A novel feature of NSI approach is that PKI Servers can exchange information with other PKI Servers. Information on certificate repositories and certificate paths is propagated in a manner similar to IP routers.

Det Norske Veritas Research is developing a **Validation Authority** (VA) for commercial use [40]. The VA approach to interoperability is an alternative to building trust structures among CAs (such as cross-signing and bridges). The VA will return a 'classification (quality indicator)' for certificates passed for validation. The implementation provides a web service based on XKMS for the relying parties.

There is work underway in the US among PKI experts in the grid research and higher education communities to establish a validation service for production use in these areas [41].

IX. Future Work

Support for dynamic certificate information is currently limited to retrieving the identifier of the issuing CA from a certificate. One of the main directions for this research will be to complete the design and implementation of the dynamic information components of the engine and other services. The validation services proposed in this paper must be developed beyond the current minimal implementation. The goal is to provide these services over standard protocols and to integrate their use into software authentication mechanisms.

Development and testing are ongoing and we hope to deploy a pilot system shortly.

X. Conclusions

A trust evaluation software system has been designed and a prototype implemented to assess the feasibility and usefulness of automatic evaluation of trust in CAs. The evaluation engine we have developed provides a good platform for experimenting with various approaches to ruleset evaluation. The experience of implementing and using the engine with the trust matrices and the initial online service suggests that this is a promising direction for our research.

References

[1] G. Pierantoni, O. Lyttleton, D. O'Callaghan, G. Quigley, E. Kenny, and B. Coghlan, "Multi-grid and multi-vo job submission based on a unified computational model," in *Cracow Grid Workshop (CGW'05)*, Cracow, Poland, November 2005.

[2] J. Astalos, R. Cecchini, B. Coghlan, R. Cowles, U. Epting, T. Genovese, J. Gomes, D. Groep, M. Gug, A. Hanushevsky, M. Helm, J. Jensen, C. Kanellopoulos, D. Kelsey, R. Marco, I. Neilson, S. Nicoud, D. O'Callaghan, D. Quesnel, I. Schaeffner, L. Shamardin, D. Skow, M. Sova, A. Wäänänen, P. Wolniewicz, and W. Xing, "International grid CA interworking, peer review and policy management through the European DataGrid certification authority coordination group," in *Advances in Grid Computing — EGC 2005*, ser. LNCS3470, P. M. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, Eds. Amsterdam, The Netherlands: Springer, February 2005, pp. 275–285.

[3] *European DataGrid project (EDG)*, 2006. [Online]. Available: http://eu-datagrid.web.cern.ch/

[4] *PKIX Charter*, IETF, 2006. [Online]. Available: http://www.ietf.org/html.charters/pkix-charter.html

[5] European Policy Management Authority for Grid Authentication in e-Science. [Online]. Available: http://www.eugridpma.org/

[6] *White Paper, Version 5.51*, eInfrastructure Reflection Group, April 2004. [Online]. Available: http://www.e-irg.org/publ/2004-Dublin-eIRG-whitepaper.pdf

[7] *Enabling Grids for E-sciencE (EGEE)*, 2006. [Online]. Available: http://www.eu-egee.org/

[8] *Distributed European Infrastructure for Supercomputing Applications*, 2004. [Online]. Available: http://www.deisa.org/

[9] *Large Hadron Collider Computing Grid Project (LCG)*, 2006. [Online]. Available: http://lcg.web.cern.ch/LCG/

[10] *South Eastern European Grid-enabled eInfrastructure Development*, 2004. [Online]. Available: http://www.see-grid.org/

[11] The Americas Grid Policy Management Authority. [Online]. Available: http://www.tagpma.org/

[12] Asia Pacific Grid Policy Management Authority. [Online]. Available: http://www.apgridpma.org/

[13] *Trust on the Grid Goes Global*, International Grid Trust Federation, October 2005. [Online]. Available: http://www.gridpma.org/docs/igtf-newsrelease-20051005.pdf

[14] *Global Grid Forum*. [Online]. Available: http://www.ggf.org/

[15] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *ACM Conference on Computers and Security*, 1998, pp. 83–91.

[16] *OpenSSL*, 2006. [Online]. Available: http://www.openssl.org/

[17] S. Henderson and J. A. Jokl, *Grids and Bridges*, University of Southern California, University of Virginia, July 2005. [Online]. Available: http://www.educause.edu/ir/library/powerpoint/PKI0509.pps

[18] *Profile for Traditional X.509 Public Key Certification Authorities with secured infrastructure (Version 4.0)*, International Grid Trust Federation, September 2005. [Online]. Available: http://www.eugridpma.org/guidelines/IGTF-AP-classic-20050930-4-0.html

[19] *Accreditation Procedures*, EU Policy Management Authority for Grid Authentication in e-Science, April 2004. [Online]. Available: http://eugridpma.org/guidelines/EUGridPMA-accreditation-20040402-1-0.pdf

[20] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, Nov. 2003, RFC 3647. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc3647.txt

[21] *WebTrust Program for Certification Authorities*, AICPA/CICA, August 2000. [Online]. Available: http://ftp.webtrust.org/webtrust_public/tpafile7-8-03fortheweb.doc

[22] *Microsoft Root Certificate Program*, Microsoft, 2006. [Online]. Available: http://www.microsoft.com/technet/archive/security/news/rootcert.mspx

[23] F. Hecker, *Mozilla CA Certificate Policy (Version 1.0)*, Mozilla Foundation, November 2005. [Online]. Available: http://www.hecker.org/mozilla/ca-certificate-policy

[24] Y. Tanaka, *APGridPMA CA Audit Checklist*, June 2005. [Online]. Available: http://forge.gridforum.org/projects/caops-wg/document/GGF14-Audit-Checklist/

[25] E. Ball, D. W. Chadwick, and A. Basden, *The Implementation of a System for Evaluating Trust in a PKI Environment*, ser. Evolaris. SpringerWein, 2003, vol. 2, pp. 263–279.

[26] R. Butler and T. Genovese, *Global Grid Forum Certificate Policy Model*, 2003. [Online]. Available: http://forge.gridforum.org/projects/ggf-editor/document/GFD-C.16/en/1

[27] J. Luna, M. Medina, and O. Manso, "Using OGRO and CertiVeR to improve OCSP validation for grids," in *GPC*, ser. Lecture Notes in Computer Science, Y.-C. Chung and J. E. Moreira, Eds., vol. 3947. Springer, 2006, pp. 12–21. [Online]. Available: http://dx.doi.org/10.1007/11745693_2

[28] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, June 1999, RFC 2560. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc2560.txt

[29] T. Freeman, R. Housley, A. Malpani, D. Cooper, and T. Polk, *Standard Certificate Validation Protocol (SCVP)*, March 2006. [Online]. Available: http://tools.ietf.org/wg/pkix/draft-ietf-pkix-scvp/

[30] P. Hallam-Baker and S. H. Mysore, *XML Key Management Specification (XKMS 2.0)*, 28 June 2005. [Online]. Available: http://www.w3.org/TR/xkms2/

[31] P. Bothner, "Kawa—compiling dynamic languages to the Java VM," in *Proceedings of the USENIX 1998 Technical Conference, FREENIX Track*. New Orleans, LA: USENIX Association, 1998, see also: http://www.gnu.org/software/kawa/. [Online]. Available: http://citeseer.csail.mit.edu/bothner98kawa.html

[32] R. Kelsey, W. Clinger, and J. Rees, "Revised^5 report on the algorithmic language Scheme," *ACM SIGPLAN Notices*, vol. 33, no. 9, pp. 26–76, 1998. [Online]. Available: http://www.schemers.org/Documents/Standards/R5RS/HTML/

[33] P. Bothner, "A Gcc-based Java implementation," in *IEEE Compcon 1997 Proceedings*, February 1997, pp. 174–178. [Online]. Available: http://citeseer.csail.mit.edu/bothner97gccbased.html

[34] O. Kiselyov, *SXML Specification Version 3.0*, 12 March 2004, see also: http://ssax.sourceforge.net/. [Online]. Available: http://okmij.org/ftp/Scheme/xml.html#SXML-spec

[35] D. W. Chadwick and A. Basden, "Evaluating trust in a public key certification authority," *Computers and Security*, vol. 20, no. 7, pp. 592–611, November 2001.

[36] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, Mar. 1997, RFC 2119. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc2119.txt

[37] J. Mason, "Filtering spam with spamassassin," in *HEANet Annual Conference 2002*, November 2002. [Online]. Available: http://spamassassin.apache.org/presentations/HEANet_2002/

[38] *New Security Infrastructure*, 2002. [Online]. Available: http://sit.sit.fraunhofer.de/_SIT-Projekte/NSI/index-en.html

[39] B. Hunter, "Simplifying PKI usage through a client-server architecture and dynamic propagation of certificate paths and repository addresses," in *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, 2–6 September 2002.

[40] J. Ølnes, "DNV VA white paper: PKI interoperability by an independent, trusted validation authority," Det Norske Veritas, Tech. Rep. 2005-0673, 6 June 2005. [Online]. Available: http://www.dnv.com/binaries/Report-2005-0673_tcm4-159646.pdf

[41] M. Helm. (2006, January) Validation service. [Online]. Available: http://www.eugridpma.org/agenda/askArchive.php?base=agenda&categ=a054&id=a054s3t8/transparencies