

Computational Techniques for Increasing PKI Policy Comprehension by Human Analysts*

Gabriel A. Weaver
Dartmouth Computer Science
Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
gweave01@cs.dartmouth.edu

Scott Rea
Dartmouth Computer Science
Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
scott.rea@dartmouth.edu

Sean W. Smith
Dartmouth Computer Science
Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
sws@cs.dartmouth.edu

ABSTRACT

Natural-language policies found in X.509 PKI describe an organization's *stated policy* as a set of requirements for trust. The widespread use of X.509 underscores the importance of understanding these requirements. Although many review processes are defined in terms of the semantic structure of these policies, human analysts are confined to working with page-oriented PDF texts. Our research accelerates PKI operations by enabling machines to translate between policy page numbers and policy reference structure. Adapting technologies supporting the analysis of Classical texts, we introduce two new tools. Our *Vertical Variance Reporter* helps analysts efficiently compare the reference structure of two policies. Our *Citation-Aware HTML* enables machines to process human-readable displays of policies in terms of this reference structure. We evaluate these contributions in terms of real-world feedback and observations from organizations that audit or accredit policies.

Categories and Subject Descriptors

D.2.1 [Software Engineering]: Methodologies;

D.2.8 [Software Engineering]: Metrics

General Terms

Management, Security, Standardization

Keywords

PKI; Certificate Policy Formalization; XML

1. INTRODUCTION

*This work was supported by the NSF (under grant CNS-0448499). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any of the sponsors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust '10, April 13-15, 2010, Gaithersburg, MD

Copyright © 2010 ACM ISBN 978-1-60558-895-7/10/04... \$10.00.

1.1 Human Analysts and PKI Policy.

Information security policies describe an organization's requirements for protecting their computational and informational assets. In X.509 Public Key Infrastructure (PKI), a natural-language *certificate policy (CP)* is a type of information security policy that documents an organization's set of requirements for trust; furthermore, a *Certification Practice Statement (CPS)* is a natural-language document that describes how the CP is implemented.

As part of the operation of PKI, human policy analysts must regularly retrieve, review and work with certificate policies and the corresponding CPS documents. Often, policy review processes (such as audits, grid accreditation, and bridging) involve comparing a policy or practice statement *under consideration* against a *trusted or accredited* one. During this process, analysts perform several operations on these natural language texts.

- **Finding and retrieving policies**, in practice, is time-consuming and tedious. For instance, in the *International Grid Trust Federation (IGTF)*, although there is a formal distribution of accredited CAs, their corresponding policies documents are not referenced in the distribution metadata. Instead, analysts must manually browse each CA's website (which isn't always listed in the metadata), locate the policy and/or practice statement, and download it.
- **Policy comparison** requires the analyst to compare sections of one policy or practice statement (e.g. "1.1," "3.2.1") with the corresponding sections in another; in theory, these sections should match, but in practice often do not (and may be missing or moved).
- **Policy transform** requires the analyst to manipulate the structure of one policy into another's reference structure (e.g., RFC 2527 or RFC 3647); again, in theory, all policies should match the RFC exactly, but in practice they do not.
- **Policy mapping** requires a combination of policy comparison and policy transform to determine the equivalency of policies and practices within two different PKIs.
- **In compliance evaluation**, the analyst examines how well issued certificates comply with relevant sections of policy. For example, do certificates that have been issued to authenticate to the grid comply with a candidate policy?

- **Content disambiguation** requires the analyst to annotate words and phrases in policy with the specific senses with which they are used. For example, ‘reasonable’ has a specific legal meaning in Dutch law but not in English law—this caused confusion among policy auditors in the *European Union Grid Policy Management Authority (EUGridPMA)*.

Currently, these review processes are done manually, taking much time and effort. An obstacle hindering all of them is the fact that the processes are all defined in terms of the underlying semantic reference structure of the policies—but human analysts are instead confined to working with the page-oriented PDF text—which may or may not match the reference structure. Auditors therefore must manually translate, in their heads, between policy page numbers and the reference structure in order to do these operations. This forces these operations to be largely manual and/or operate on the entire document. Figure 1 sketches this situation.

1.2 Our Vision

Our overarching research vision is to accelerate PKI policy operations by building automated tools to eliminate slow and error-prone manual processes. In addition to our team’s real-world PKI operations experience, we also bring a secret weapon: experience in building automated tools to assist *classics scholars* in overcoming a similar obstacle: doing semantic analysis on page-navigable reference works [20]. (In this earlier paper, we helped apply simple clustering algorithms and text-mining techniques to empirically illustrate how Homeric scholia (scholarly comments written in manuscripts) were transmitted, arguably rewriting the past 200 years of theory regarding their transmission.)

As a first step towards achieving this vision, we applied the *Canonical Text Services (CTS) Protocol* (a tool we used in classics work [19]) to construct the *PKI Policy Repository* [18]. Our *PKI Policy Repository* solved the policy retrieval problem. Before, analysts had to manually find and then browse each CA’s website. Using the repository, analysts request an arbitrary fragment of policy, the request is encoded as a CTS-URN [10] (a hierarchical, machine-actionable, human readable reference string), and the appropriate passage is retrieved. Using this machine-actionable reference framework, we reduced the time to aggregate data for CP comparison by up to 94% (*Policy Reporter*) and reduced the time to map policies from hours to seconds (*Policy Mapper*).

In this current paper, we report on further progress in achieving this research vision. In particular, we focus on the *human-computer semantic gap* between the machine representation of PKI policies (structured by page) and the ways in which policy analysts interact with policy (structured by reference scheme). We contribute tools and techniques that use computation to help analysts efficiently compare and browse policies:

- Our *Vertical Variance Reporter* computes and reports differences in the reference structure of two policies.
- Our *Citation-Aware HTML* enables machines to search, to style, and to process human-readable displays of policy in terms of this reference structure.

We also discuss the tools we plan to build next in order to complete the vision.

These tools, in combination with our prior work, provide better quality, reproducible, and reliable data upon which policy auditors can base their trust decisions. Figure 2 sketches how we envision these contributions transforming PKI policy operations.

1.3 This Paper

In Section 2 we describe a set of principles and technologies from the Classics that directly inform our research on PKI policy. Section 3 presents motivation: real-world feedback and observations from organizations—like the FPKIPA-CPWG, EuGridPMA, and TAGPMA—that audit or accredit policies. In Section 4 we describe the design and implementation of our *Vertical Variance Reporter* and *Citation-Aware HTML*—and also discuss the next tools we plan to build. Section 5 gives an experimental evaluation of our *Vertical Variance Reporter* and describes the design of several applications that leverage the properties of our *Citation-Aware HTML*. Section 6 reviews relevant work. Section 7 describes future research directions building upon this work, and Section 8 concludes.

2. MAPPING CLASSICAL TECHNOLOGIES TO PKI

Our work adapts technologies from the Classics to construct computational tools that accelerate traditionally, exclusively-manual PKI policy operations. PKI policies are reference works. Analysts need to be able to align policy sections for comparison. Section 5 of RFC 2527 and Section 6 of RFC 3647 effectively define a canonical structure for *Certificate Policies (CP)* and *Certification Practices Statements (CPS)* for authors and users to understand the meaning and scope of these texts.

Traditionally, PKI policy operations require analysts to manually align policy sections for comparison. However, we can regard these natural language texts as reference works, with canonical structures for authors and users to understand the meaning and scope of these texts. (e.g., Section 5 of RFC 2527 and Section 6 of RFC 3647 define the structure for *Certificate Policies (CP)* and *Certification Practices Statements (CPS)*.)

Prior work in the classics (to which we contributed, in fact) provides technologies to help with analogous tasks for the natural language texts that field studies. We can build on these technologies to solve our PKI problem. In this section, we review some principal building blocks the Classics gives us:

- a data model for canonical texts
- a historical distinction between physical navigation and logical reference, and
- a methodology for working with multiple editions of the same work.

2.1 A Data Model for Canonically Cited Texts

Both theoretical work and hands-on experience with digital texts in the Classics (e.g. Homer and Archimedes [17]) over the past twenty years [11] [9] led us to propose in our previous Classics work [20] that all canonically cited texts possess four properties:

1. citable units of a text are ordered

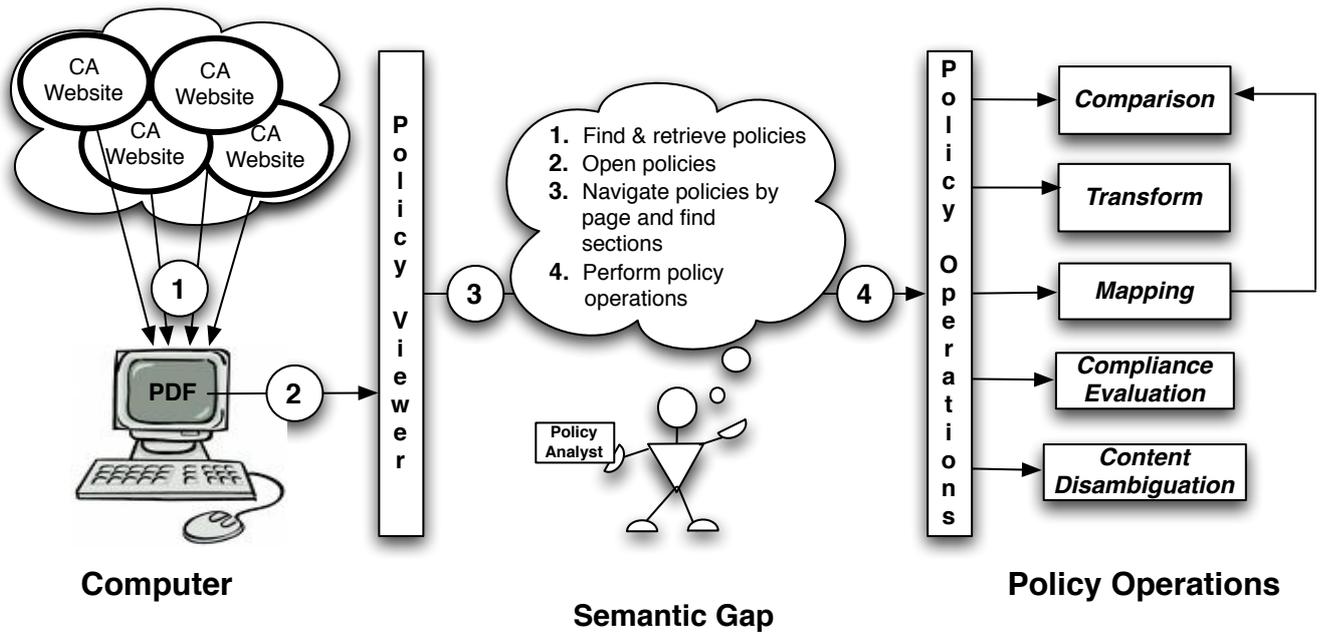


Figure 1: Policy analysts operate on PKI policies by their reference structure, but machine representations of policy like PDF are organized by page. This imposes a semantic gap, forcing policy operations to be largely manual.

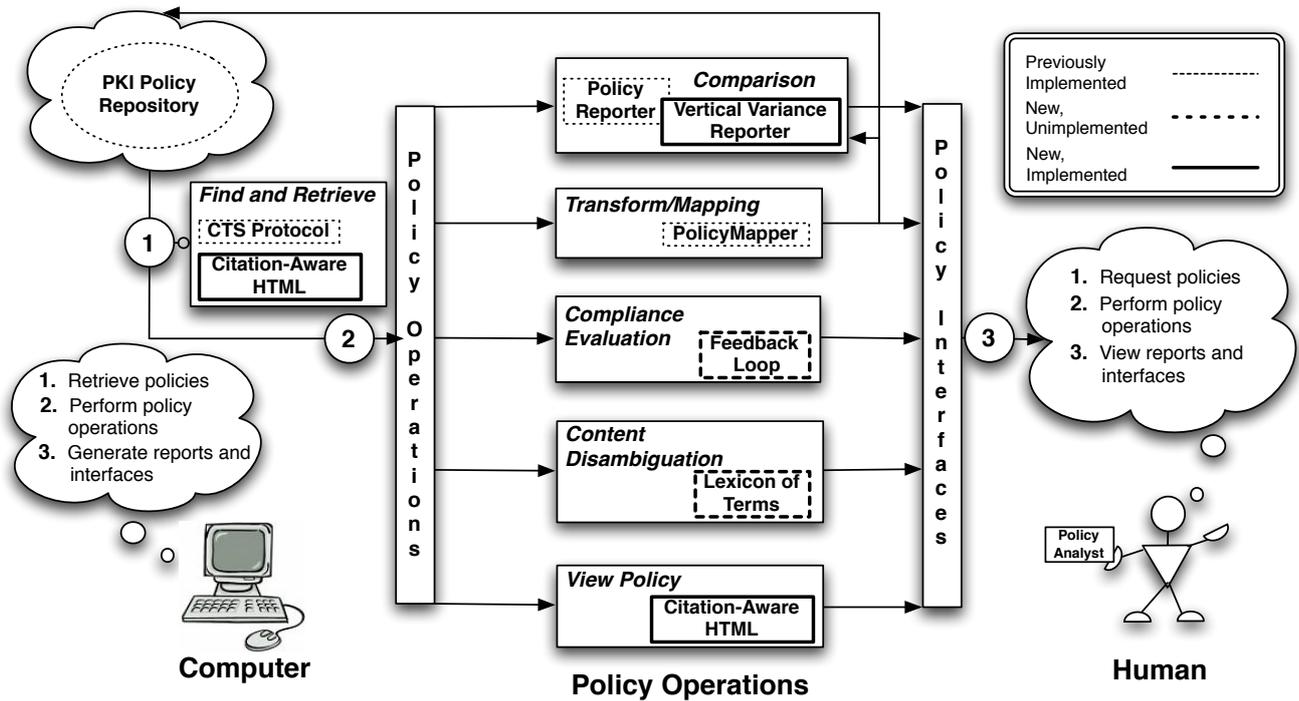


Figure 2: Our representation of policy allows man and machine to directly operate on PKI policy, resulting in reproducible, more reliable data for helping analysts make policy decisions. Please note that previously implemented items were either built or adapted by us.

2. citable units of a text are organized in a (possibly flat) hierarchy
3. versions of a text are related to a notional text in a conceptual hierarchy
4. citable units may include mixed content

The *Canonical Text Services (CTS)* library encodes this data model for canonical texts. Our CTS Protocol [19] defines an HTTP protocol in terms of this data model for referencing and retrieving arbitrary passages of a text.

Our initial work applying Classics tools to PKI contributed the *PKI Policy Repository*, consisting of a CTS server loaded with validated, XML PKI policies. We encoded PKI policies using *Text Encoding Initiative (TEI) P5 Lite*, an XML standard for representing texts in digital form [2]. Like previous efforts to encode policies using XML [5] [4], we modeled a security policy as a tree. This tree corresponded directly to both the hierarchy in the second property of our data model for canonically cited texts and the outline of provisions in Section 5 of RFC 2527 [7] and Section 6 of RFC 3647 [8]. Given a policy’s text, we only mark up this hierarchical *reference structure*. By keeping the markup light, we reduce the complexity of encoding a policy.

2.2 Physical Navigation and Logical Reference

The Classics also teaches us the important distinction between physical navigation and logical reference. Originally, when texts such as Homer appeared on manuscripts (MSS), one could reference individual books or lines of the poem, but resolving the reference to a passage of text required manually flipping through the physical MSS folios. With the arrival of the *book* (as opposed to manuscript), the page number and table of contents enabled scholars to quickly resolve logical references (such as “Book 9 of the *Odyssey*”) to physical pages for that particular printing. However, over time these tools for physical navigation were used as a citation mechanism [15]. Disciplines outside of the Classics and law, who stuck with logical citation schemes, began citing works in terms of the page. For examples, professors who reference pages rather than logical sections in their syllabi must update their syllabus if the textbook edition or printing changes. CTS advances the historical evolution of text, enabling people and processes to retrieve and navigate texts by their logical structure.

Once policy analysts can use computers to retrieve passages by logical citation, they are no longer required to manually translate, in their heads, between policy page numbers and the reference structure used by many policy operations. In actual practice policies are represented as untagged PDFs that are structured according to the page. Even services such as Google books do not allow one to explicitly retrieve or search within a specific section of a text.

Our overall research vision frees the analyst to continually work in logical reference coordinates whether retrieving, comparing, or mapping a certificate policy. Translation from these logical coordinates to a physical coordinate scheme (byte offsets in a file) is outsourced to the computer. Since the computer can perform this translation, many policy operations can also be augmented with computational tools.

2.3 Working with Multiple Editions

Combining the above properties of canonically cited texts with a citation by logical reference provides Classical scholars with a framework to analyze multiple editions of a text. Versions of a text are related to a notional text (the work) in a conceptual hierarchy. For example, the various translations and editions of Homer’s *Odyssey* can be viewed as descendants of a notional work. Although versions may differ, they share (more or less) a common logical reference structure. Book 9 of the *Odyssey* contains Odysseus’ adventures with the cyclops Polyphemus regardless of the edition or translation.

Classical scholars also realized that editions may contain slight variations both in logical reference structure, and in textual content. To address these problems, Nagy introduced the concepts of *vertical variance* and *horizontal variance*, distinguishing between differences in structure and content respectively [14].

In PKI operations, we can view the RFC 2527 and RFC 3647 policy formats as notional works according to which individual CAs author editions. Like Classical scholars, policy analysts analyze multiple editions of a text using a common set of logical reference coordinates. Furthermore, different editions may differ in terms of structure or textual content. Like passages in Homer, PKI policy sections may be added or deleted over time. Unlike Homer however, PKI policy passages are identified not just by passage reference (e.g., “(9)”) but also by headers that describe the purpose of the section (e.g., “Other Business and Legal Matters”). Therefore, passage reference does not necessarily correlate with section semantics. (This would be like Polyphemus the cyclops occurring in Book 6 rather than Book 9 of the *Odyssey*!) Headers may be relocated and paired with a different passage reference, identifying a different but semantically equivalent section to the corresponding section in the canonical reference structure.

To address these problems in PKI, we developed the *Vertical Variance Reporter* to compute and report vertical variance between multiple editions of a policy under these conditions, enabling policy analysts to see the mapping between two policies’ reference structures.

3. REAL-WORLD MOTIVATION

3.1 Feedback

In our prior PKI policy tool work, we developed the *PKI Policy Repository*, *Policy Reporter*, and *Policy Builder*. When we presented these tools to the FPKIPA-CPWG, EuGridPMA, and TAGPMA, these organizations gave us feedback.

Many analysts agreed that a policy repository was desirable for finding policies, understanding the *actual* content of real-world policies, and dynamically creating new policies from previously-accredited, well-understood policies. However, they cited three major obstacles preventing the adoption of our approach: *encoding speed*, *policy variation*, and *display quality*. This current paper contributes solutions to the last two concerns as part of a larger strategy to increase encoding speed—and discusses our plan to eliminate the remaining obstacle.

- *Encoding Speed*. Based upon our prior evaluation of the *Policy Reporter*, we could encode a policy in 4-6 hours by copying and pasting policy content from a PDF into a TEI-XML file.

- *Policy Variation.* Once a policy was encoded and loaded into the *PKI Policy Repository*, analysts could retrieve and run analyses on multiple editions of one or more policy sections, expressed as a set of passage references. However, this approach implicitly assumed that passage reference correlate to section semantics. In the real-world, headers may be relocated and paired with a different passage reference, identifying a different but semantically-equivalent section to that listed in RFC 2527 or 3647. Analysts urged us to generalize our approach to handle the relocation of headers.
- *Display Quality.* Our *PKI Policy Repository* is primarily a service for computer programs; analysts wanted a more human-friendly display of our XML policies. Paragraphs, images, and tables needed to be clearly displayed. Although analysts saw the potential of augmenting their policy operations with computational tools, they required a way to view the XML policy using the traditional typographical conventions that reflect policy structure (for example, using different sized fonts to denote sections and subsections of a policy).

3.2 Observations

In addition to gaining feedback from our work, attending meetings of these accrediting organizations allowed us to directly observe presentations, discussions, and business procedures which would benefit from our computational framework once it could accommodate vertical variance and provide a better human interface for browsing policies.

Policy analysts *manually align* policy provisions before they can compare their content. However, the real world makes this task harder than one expects. Sometimes a policy under consideration contains additional sections that do not map to the trusted or accredited policy. Furthermore, such non-standard sections may contradict statements made in other, standard sections of policy (analysts at the FPKIPA-CPWG call this the *whitespace problem*). Such contradictions, if present in an accredited policy, increase the risk accepted by an accrediting organization. However, a tool that measured the vertical variance of a policy would allow analysts to quickly identify non-standard sections of a candidate policy where these contradictions are likely to occur.

Analysts' current approaches to finding, searching, annotating, and evaluating policies could be accelerated with better human interfaces for browsing policies. Although the IGTF provides a formal distribution of accredited CAs, the corresponding policies themselves are not referenced in the distribution metadata. Analysts searching for terms over the entire text of a PDF policy complained that one could not restrict the search space to a particular section or range of sections. Analysts manually generate matrices consisting of policy sections and comments—so a framework that supported annotation of policy would allow them to dynamically generate these comparison matrices.

Researchers at Trinity College, Dublin presented a suite of unit tests for measuring the validity of a certificate relative to a policy [3]; we saw the potential for combining these automated tools with our suite of policy creation and analysis tools for allowing policy analysts, both non-technical and technically-inclined, to experiment with how modifying a policy's text impacts certificate validity.

4. OUR COMPUTATIONAL TOOLS

As noted above, the policy analysts at the FPKIPA-CPWG, EUGridPMA, and TAGPMA cited three major obstacles to our prior contribution: *encoding speed*, *policy variation*, and *display quality*. We now discuss the tools we built (and the tools are still building) to address these obstacles—and further manual bottlenecks we perceive.

4.1 Completed Tools

4.1.1 Vertical Variance Reporter

Our *Vertical Variance Reporter* addresses the practitioner community's concern over *policy variation*.

In order to determine the *actual* reference structure of a policy rather than imposing an *idealized, trusted structure* such as RFC 2527 or RFC 3647, we extract section identifiers (passage references and their corresponding headers) from its table of contents. Parsing relies upon a library of regular expressions we built to parse common formats for tables of contents. Iterating through these sections, we output a list of section identifiers for the *Vertical Variance Reporter*.

Our *Vertical Variance Reporter* takes two lists of section identifiers as input and computes a mapping between the two that preserves semantic-equivalence. Think about the section identifiers in the *policy under consideration* as being mapped, by some unknown function, to the section identifiers in the accredited policy. We want a way to automatically discover and then calculate this function (or at least a good approximation thereof; the human can do the rest).

To do this, we use one of the secret weapons inspired by the Classical notion of vertical variance: a *confusion matrix* built using the *Levenshtein* metric for semantic distance.¹ The *Vertical Variance Reporter* first records the distance between section headers in the source and target policies. Our tool then processes the confusion matrix to report a bidirectional mapping, classifying policy sections as matched, relocated, or unmapped.² In the next few paragraphs, we provide more details about how we compute the confusion matrix and then use it to infer a mapping.

We use a confusion matrix to (1) detect passage references in the *trusted or accredited policy* that are missing from the *policy under consideration*, (2) identify sections in the policy under consideration whose headers are within epsilon of a section header (via the Levenshtein distance) from the accredited policy, and (3) identify sections in the policy under consideration which are further than epsilon away from any of the target policy headers. The rows of the confusion matrix are indexed by the *possible* passage references within source policy given the target. These index values directly correspond to the passage references in the target policy which are used to index columns.

Our tool computes the confusion matrix by iterating over each of the passage references in the target policy and first testing whether it is enumerated in the source policy section list. If the target passage reference does not appear in the source list, a -1 is recorded in the confusion matrix for the entire row. If the source section list does contain the target passage reference, then we calculate the Levenshtein distance between the target header for the current target passage reference and each of the headers in the source. Re-

¹We use the Levenshtein distance but another metric could be used instead.

²It should be noted that this technique may prove useful in clustering documents based upon their reference structure.

sults are recorded in a two dimensional matrix where rows correspond to *possible* passage references within a source policy given the target policy and columns correspond to the target policy’s passage references.

The *Vertical Variance Reporter* infers a mapping from two confusion matrices, one comparing sections in the source to those in the target, the other comparing sections in the target to those in the source. In this way, we obtain (1) a list of omitted target references, (2) a list of matched source headers (identified by passage reference), and (3) a list of unmatched source headers. From the target-to-source matrix, we obtain a list of additional source references, a list of matched target headers, and a list of unmatched target headers. By processing these lists our tool is able to classify a section as mapped or unmapped. Mapped sections may be exact matches where the passage references in source and target are equal and the Levenshtein distance is 1, fuzzy matches where the passage references may be different or (inclusive) the Levenshtein distance exceeds a threshold (we used 0.90). Source sections may be unmapped because their passage reference is not present in the target document and their headers fail to match (additional sections) or simply because their headers failed to match any of the target headers (unmatched sections). Table 1 (located at the end of this paper) shows and discusses excerpts of reports generated by our *Vertical Variance Reporter*.

4.1.2 Citation-Aware HTML

In order to address the practitioner community’s concern over *display quality*, we developed *Citation-Aware HTML*, which makes it possible for human analysts to search, to style, and in general to manipulate policy in the browser according to logical reference,

Given a list of section identifiers, we use Lucene [12] to index and search Google’s OCR HTML for the corresponding byte offset at which the section begins.³ Our HTML generation process then iterates through these locations, extracting the textual content contained between the start of the section and the next successfully-translated section (or end of file).

Citation-Aware HTML classifies HTML elements using CTS-URNs via the *class* attribute and thereby relates the content spanned by those elements to a policy’s reference scheme via machine-actionable reference. Our *Citation-Aware HTML*, like TEI-XML representations of policy, encodes the hierarchy of citable units within a policy. An important consequence of this is that the mapping of *citation nodes* (citable units represented by the *Document Object Model, DOM*) between TEI-XML and HTML is bijective: changes to any citation node in either format can be mirrored in the other since one can generate either format by processing the other.

Our *Citation-Aware HTML* format allows humans to view text using traditional typographical conventions that reflect policy structure while gaining the benefits of navigation by logical reference. Although this technique could be applied to any HTML document, parsing Google’s OCR allows us to extract CSS styling information so that eventually we can maintain the typographical conventions in the original PDF policy. This will allow us to faithfully reproduce the *display* of paragraphs, lists, and tables and may be useful

³Note that we are using Lucene to translate a logical reference coordinate system to a physical coordinate system (bytes) for our machine representation (HTML file).

for their eventual *encoding* in TEI-XML. Furthermore, our technique lends itself to several policy-browsing applications whose design we discuss below.

4.2 Tools Still Under Development

4.2.1 Policy Encoding Toolchain

We are addressing the practitioner community’s concern over *encoding speed* with our *Policy Encoding Toolchain*. Encoding a PDF policy with our *Policy Encoding Toolchain* requires the following three steps: (1) use Google Docs to generate Google’s OCR HTML output for a given PDF policy, (2) parse this HTML to generate a TEI-XML encoding as well as CSS styling information, and (3) generate a high-quality, human-readable view of the policy that faithfully recreates the typography seen in Google’s OCR HTML.

Extracting section lists from a policy’s table of contents as well as generating *Citation-Aware HTML* are both components of our toolchain that have value in and of themselves. In order to generate TEI-XML from Google’s HTML, we must be able to generate a list of sections describing the reference structure we are trying to represent. Our *Vertical Variance Reporter* compares the vertical variance of two policies, allowing us to evaluate the quality of the encoding of a policy using a given list of section headers. However, this same tool is also useful to policy analysts in comparing a *policy under consideration* to a *trusted or accredited policy*. Our *Citation-Aware HTML* is a product of our envisioned toolchain. However, this same format has independent utility as a key component of several of our policy browsing applications which we will now describe.

4.2.2 Policy Browsing

Policy-browsing applications based upon our *Citation-Aware HTML* include a search utility for finding policies or searching within arbitrary sections of policy, a policy annotation framework generalizing the idea of using typographical cues (font size, color, etc) to reflect policy structure, and a policy feedback loop for dynamic certificate validation which relies upon the bijective mapping between HTML and TEI-XML.

Citation-Aware Searching.

Since the *class* attribute of each citation node is annotated with its corresponding CTS-URN, search engines that index *Citation-Aware HTML* should, in theory, be CTS-URN aware. This means that one could search for all IGTF policies, all policies from a particular CA, a particular version of a policy, or a particular passage of a policy by searching for a particular CTS-URN. At the very least, retrieval of a particular edition should be possible since *Citation-Aware HTML* contains a URN in its page metadata. Just as one can use geographic coordinates to restrict a search to a particular region, so can one use CTS-URNs as textual coordinates to restrict a search to a particular region of text.

Policy Annotation Framework.

Although Google’s OCR HTML styles content to mimic page typography, for applications like annotating policy, our *Citation-Aware HTML* enables one to style content with respect to its reference scheme. For example, auditors could highlight various policy sections to indicate the presence of an annotation.⁴ Alternatively, auditors could just color-code

⁴These annotations could be mined and presented in a matrix.

policy sections to indicate the various levels of compliance or issues that need further review.

Policy Feedback Loop.

Our *Policy-Driven Feedback Loop* allows analysts to empirically explore the effect that changing a policy would have on an actual PKI infrastructure. Figure 3 illustrates our design that would enable policy analysts to iteratively evaluate the effects of changing policy on certificate validity. First, policy analysts issue a request for a passage of policy against which to check the validity of a corpus of certificates. Using a CTS *GetPassage* request, the corresponding TEI-XML is retrieved and used to generate a suite of unit tests. The test results are then presented by controlling the styling of our *Citation-Aware HTML* for the requested policy passage. For example, the RFC 2119 significance level of violated policy assertions could be indicated with different colors, the number of certificates failing to comply with an assertion could be indicated by font size. Policy writers could then adjust the required value or significance of a policy assertion and POST the updated HTML. Since the mapping between TEI-XML and HTML citation nodes is bijective we can construct a feedback loop: the HTML citation nodes can be used to recover the XML. New unit tests can then be generated and new results presented back to the analyst.

The feedback loop depends upon enriching the reference model for policy with assertions on certificate content. Rather than hand-coding unit tests for *every* new version of a policy, we hand tag the expected value, relation, and significance of each machine-enforceable policy statement *once* within the TEI-XML. Our previously-developed *RFC 2119 analysis* tool leveraged the well-defined semantics of MUST, SHALL, and OPTIONAL. Since these words are technical terms, we were able to process occurrences of these words as tokens with a specific meaning. Similarly, by enriching our reference model with a representation for assertions on certificate content, we hope to gradually develop a lexicon of technical terms for disambiguating content and gradually make larger and larger portions of human-readable policy machine-actionable.

Using our extended policy representation, we walk the tree of citation nodes of the requested policy passage and generate a unit-test suite, much as a compiler walks an *Abstract Syntax Tree (AST)*. The expected value, relation, and significance encoded by our model of assertions, are treated as parameters for generating each unit test. Each citable assertion results in the generation of a unit test whose name encodes its corresponding citation node and significance. The unit tests are executed, results interpreted, and used to generate a CSS style to be included in the *Citation-Aware HTML* for the requested passage. Policy analysts may change the values in the assertions, choosing terms from a *controlled vocabulary* derived from our lexicon.

5. EVALUATION

In this section we present empirical and anecdotal evidence to argue that our *Vertical Variance Reporter* and *Citation-Aware HTML* tools satisfy many of the requirements inspired by feedback and observations from real-world policy analysts. (As noted earlier, our other tools are still in development.)

5.1 Vertical Variance Reporter

The *Vertical Variance Reporter* addresses the need to be able to understand how the structure of policies differs so that one can quickly determine which sections of a *policy under consideration* can be compared to an *accredited or trusted policy*. In this section, we discuss results from experimental evaluations of how the section identifier extraction process affects the ability to infer a policy mapping between source and target policies. During the discussion of results, we will also mention how this tool relates to the feedback and observations from real-world policy analysts.

5.1.1 Parsing Sections from Tables of Contents

The *Vertical Variance Reporter* computes a semantics-preserving mapping between two lists of section identifiers. Our main technique for generating these lists is to parse the table of contents for a policy in Google's OCR HTML output. In order to make claims on how well the reference structure described in a policy's table of contents (TOC) maps to a target reference structure (such as RFC 3647), we need to be sure that we can correctly extract section identifiers from table of contents formatted in Google's OCR HTML. In the first evaluation, we chose 10 policies, generated Google's OCR HTML, extracted their tables of contents, and parsed them for section identifiers. (As noted earlier, we are currently building a tool to automate this encoding process.)

Table 2 shows results for the final step: parsing section identifiers from tables of contents.

As one can see, parsing the table of contents of these policies takes only seconds and we successfully extract every header contained therein. It should be noted that the extracted headers may contain minor artifacts from the extraction process such as rogue page numbers and page headers. These artifacts can be easily fixed either with some quick manual editing or global find and replace. The results of Evaluation 1 allow us to say that our section lists, accurately reflect the policy structure described in a policy's table of contents.

5.1.2 Computing Vertical Variance Using Tables of Contents

The second evaluation uses our *Vertical Variance Reporter* to compute the vertical variance between the same 10 source policies and the structure of RFC 2527 or RFC 3647 depending upon the source policy. We use the section lists derived from the tables of contents. This evaluation allows us to see how well the documented structure of a source policy maps to the RFC standard. Results are presented in Table 3.

Looking at the results we see that the AustrianGrid table of contents' closely follows RFC 3647 (containing 267 of the 270 RFC sections) while the TACC Root policy appears to be missing many sections (containing 67 of those 270 sections). Looking at the ULAGrid policy we see that it contains 271 citable units whereas RFC 3647 only contains 270. This indicates an additional section which the report will identify. This kind of information is a useful first step for solving the *whitespace problem*; it identifies sections to policy analysts that are non-standard and therefore may contain potentially contradictory information. Our mapping from the Austrian Grid TOC to RFC 3647 shows that 260 out of 267 citable units were successfully mapped and that the other 7 units were classified as unmapped. Only 65 of the already-reduced 67 sections in the table of contents for

Dynamic Policy Evaluation

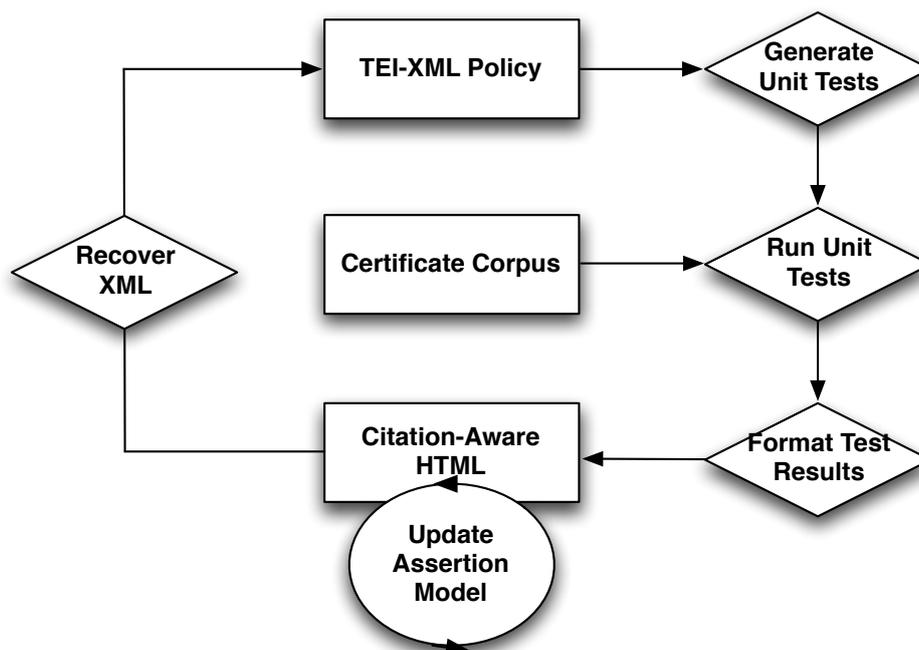


Figure 3: Dynamic Policy Evaluation will allow the policy analyst to treat *Citation-Aware HTML* policies as a form for configuring a certificate policy validation engine. Results of testing the modified policy against a corpus of certificates will be highlighted within the submitted text according to degree of compliance and significance of policy assertion.

TACC Root, actually corresponded to sections seen in RFC 3647. Notice that the mapping from RFC 3647 to Austrian Grid is consistent with its inverse, indicating that we are mapping the same 260 citable units in both directions.

5.1.3 Computing Vertical Variance Using Enhanced Section Lists

Evaluation 3 uses additional sources of information to increase the size of the source section list which we will refer to as *TOC+*. Increasing our section lists is necessary since the tables of contents of some policies do not contain all of the sections *actually* contained in the policy. In Table 3, we see that the DFN-PKI 2.2 policy only contains 79 out of 270 possible sections from RFC 3647. However, looking at the policy text, one sees several sections which its table of contents does not enumerate. Because of this, we paired unmatched passage references from Evaluation 2 with section headers from the target policy, searched for them within our source policy, and if the search returned a unique hit, folded them into our source section header list. Table 3 shows results of this experiment.

Looking at the results, we see that in some cases, this technique increased the size of the enhanced section lists (*TOC+*). DFN-PKI 2.2 went from having 79 citable units to 203 citable units. TACC-MICS' policy went from 151 citable units to 270 citable units. This was because TACC-MICS' policy did not enumerate level 3 citation nodes (e.g. "1.3.2") but only levels 1 and 2 (e.g. "1", "1.3" respectively). Many of these newly-inventoried sections could be resolved to an RFC 3647 section: 200 of the 203 citation nodes in the DFN-PKI 2.2 policy could be mapped to RFC 3647. However, some policies did not benefit at all from this approach,

the TACC Root policy, with only 67 sections inventoried remained unchanged. On the flip side of the coin, the Austrian Grid policy, with only 3 fewer sections than that of RFC 3647 also remained unchanged. It should be noted that in general, inferring all mappings took between 9 and 45 seconds. Generating enhanced section lists took between 8 and 76 seconds depending upon the size of the section list to be augmented. We ran our evaluations on a MacBook Pro running MacOS 10.5 on a 2.33 GHz Intel Core 2 Duo processor and 2 GB 667 MHz of DDR2 SDRAM.

5.1.4 Comparing Enhanced Section Lists to Ground Truth

Evaluation 4 uses a ground-truth list of policy headers to generate results as in Evaluations 2 and 3. We manually went through each policy and compiled a list of headers in the actual CP or CPS. We then ran the *Vertical Variance Reporter* to infer a mapping between our ground truth lists (*GroundTruth*) and our enhanced section header lists, allowing us to quantify how well we approximate actual policy structure. Table 4 shows results of this experiment.

Our results in Table 5 indicate that headers extracted using our enhanced section list methodology (*TOC+*) approximated the actual structure of policies in our corpus with 90.9% to 100% accuracy. Most policies follow the standard format described in RFC 2527 and RFC 3647. The FBCA CP was an exception as it contained 28 non-standard provisions with citation depth 4. For example, Section 6.2.3.4 is found in FBCA CP but is not found in RFC 3647. If one considers only provisions between depths 1-3 inclusive, then we successfully identify between 97.8%

and 100% of all actual provisions. Furthermore, we were able to map our $|TOC + |$ headers to 89.0% to 99.6% of all *GroundTruth* headers.

5.2 Citation-Aware HTML

As discussed earlier, we developed *Citation-Aware HTML* in direct response to real-world feedback on our *PKI Policy Repository*. In direct feedback, analysts wanted a human-friendly display of XML policies with paragraphs, images, and tables within the policies preserved and presented. In observing policy organizations, we also saw the potential to use better human interfaces for browsing policies to accelerate and improve the process of searching, annotating, and evaluating policies.

5.2.1 Addressing Feedback

Our *Citation-Aware HTML* gives policy analysts a more human-friendly display of XML policies with the potential to exactly replicate the presentational results of Google's OCR output. Currently, we have a basic algorithm for encoding paragraphs. Given that Google does not display embedded images or explicitly *encode* tables in their OCR output, we will hand code image references. The *display* of paragraph, lists and tables will be preserved through styling information which we extract from Google's OCR. However, should individual rows or cells of a table need to be referenced and retrieved by machine, then hand coding their semantic structure within the TEI-XML will become necessary. It should be noted in spite of these limitations, we expect that using our *Policy Encoding Toolchain* to generate XML for most of the policy combined with manual encoding of images and tables as needed, will significantly reduce policy encoding speed.

5.2.2 Leveraging Observations:

Our design descriptions for *Policy-Aware Searching*, a *Policy Annotation Framework*, and a *Policy Feedback Loop for Certificate Validation* all rely upon key properties of *Citation-Aware HTML* to help analysts search, annotate, and evaluate policies. First, we classify citation nodes in the HTML with CTS-URNs, a reference string whose semantics are well-understood and that machines can process, whether to index content for searching or style content according to some meaningful convention. Secondly, we leverage the second fundamental property of canonically cited texts to realize that the mapping between citation nodes in TEI-XML and HTML is bijective. This allows us to create a dynamic *policy feedback loop* that technical and non-technical policy analysts can use to dynamically evaluate the consequences of changes in policy.

6. RELATED WORK

Semantic HTML and Semantic CSS advocates write HTML and CSS that emphasizes the meaning of the text over its presentation [13]. Our *Citation-Aware HTML* subscribes to this philosophy but goes further by embedding URNs to associate semantics with page content. Additionally, others have recommended using Google OCR to convert PDF files into text [1].

The *Policy-Driven Feedback Loop* directly builds upon work done by David O'Callaghan at Trinity College, Dublin [3]. His work will provide us with target and source languages for our policy assertion to unit test compiler. Inglesant, Chad-

wick, and Sasse developed a controlled vocabulary for configuring access control policies expressed in XML [6]. Our work takes a similar approach, encoding select portions of natural language PKI policies, and deriving a controlled vocabulary from a lexicon of observed words and phrases.

Our work builds upon established standards and mature technologies. TEI P5 [2] represents 15 years of research in encoding texts with XML. The CTS Protocol [19] has been in development for 5 years and is based upon over 20 years of experience [9] in computing with a variety of digitized texts.⁵

7. FUTURE WORK

Using our tools to quantify vertical variance and browse policy in terms of its underlying structure, we will build an IGTF PKI Repository based upon the policies in its distribution. Using confusion matrices we will quantify the structural variance in the IGTF's policies. Knowing which sections of policy are semantically comparable, we will then be able to quantify their horizontal variance.

Two approaches we will employ in quantifying horizontal variance include adding structure to our TEI-XML editions of policy, and using text mining, much as we did in [20], to identify patterns in content with respect to a text's structure. Extending our markup with other data structures, such as assertions, represents a general approach. Most people roughly agree upon the reference structure of a policy. The data models arising from interpreting the text varies greatly. We intend to continue to make content machine-actionable by extending our markup to include structures of interest and to document content values in a machine-actionable lexicon. However, our approach also enables us to use textual content alone to extract topics relevant to trust decisions. With the IGTF repository, we will train classifiers to find all information in a document relevant to a topic. This is of special interest to the FPKIPA-CPWG.

8. CONCLUSION

The *Vertical Variance Reporter* and *Citation-Aware HTML* are our solutions to challenges posed by real-world policy reviewers all over the world. Our *Vertical Variance Reporter* allows analysts to quickly compare the reference structures of two policies and find semantically-equivalent sections between them. Our *Citation-Aware HTML* not only gives policy analysts a nicely-formatted view of policy but also allows us to create a variety of applications for searching, annotating, and evaluating policy. By aligning the textual coordinate systems of man and machine, we have narrowed the *human-computer security policy gap*. Given that human-judgement alone can actually weaken the effects of a security policy [16], we intend to continue exploring how computational tools can support human judgements in the analysis and enforcement of security policy.

9. REFERENCES

- [1] Amit Agarwal. Perform OCR with Google Docs and Turn Images Into Editable Documents. Retrieved on November 20, 2009 from <http://www.labnol.org/internet/perform-ocr-with-google-docs/10059/>.

⁵We used this experience in designing the CTS Protocol, requiring compatibility with texts encoded in TEI, DocBook, or any other valid XML format encoding a citation scheme.

- [2] L. Burnard and S. Bauman. TEI P5: Guidelines for electronic text encoding and interchange. *Text Encoding Initiative Consortium*. Retrieved July, 11:2008, 2007.
- [3] David O' Callaghan. Automated Certificate Checks, 2009.
- [4] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak. An Innovative Policy-Based Cross Certification Methodology for Public Key Infrastructures. In *EuroPKI*, 2005.
- [5] V. Casola, A. Mazzeo, N. Mazzocca, and V. Vittorini. Policy Formalization to Combine Separate Systems into Larger Connected Network of Trust. In *Net-Con*, page 425, 2002.
- [6] David W. Chadwick and A. Sasse. The Virtuous Circle of Expressing Authorization Policies. In *Semantic Web Policy Workshop*, 2006.
- [7] S. Chokhani and W. Ford. RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, March 1999.
- [8] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu. RFC 3657: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, November 2003.
- [9] Gregory Crane. The Perseus Digital Library. Retrieved May 29, 2009 from <http://www.perseus.tufts.edu/hopper/>.
- [10] D.Smith. CTS-URNs: Overview, December 2008. Retrieved May 29, 2009 from <http://chs75.harvard.edu/projects/diginc/techpub/cts-urn-overview>.
- [11] C. Dué, M. Ebbott, C. Blackwell, and D. Smith. The Homer Multitext Project, 2007. Retrieved May 29, 2009 from http://chs.harvard.edu/chs/homer_multitext.
- [12] Welcome to Lucene! Retrieved November 20, 2009 from <http://lucene.apache.org/>.
- [13] Antonio Lupetti. CSS coding: semantic approach in naming convention. Retrieved on November 20, 2009 from <http://woork.blogspot.com/2008/11/css-coding-semantic-approach-in-namin%g.html>.
- [14] Gregory Nagy. Editing the Text: West's Iliad. *Homer's Text and Language*, pages 54–56, 2004.
- [15] L.D. Reynolds and N.G. Wilson. *Scribes and scholars*. Clarendon Press, 1967.
- [16] Stephanie A. Trudeau, Sara Sinclair, and Sean Smith. The Effects of Introspection on Creating Privacy Policy. In *Workshop on Privacy in the Electronic Society*, 2009.
- [17] G. Weaver. Semantic and Visual Encoding of Diagrams. Technical Report TR2009-654, Dartmouth College, Computer Science, Hanover, NH, August 2009.
- [18] G. Weaver, S. Rea, and S. Smith. A Computational Framework for Certificate Policy Operations. In *Public Key Infrastructure: EuroPKI 2009*. Springer-Verlag LNCS., 2009. To appear.
- [19] G. Weaver and D. Smith. Canonical Text Services (CTS). Retrieved May 29, 2009 from <http://cts3.sourceforge.net/>.
- [20] G. Weaver and D. Smith. Applying Domain Knowledge from Structured Citation Formats to Text

Mapping	AustrianGrid Ref	Section Class	(3647 Ref, Score)
$S \rightarrow T$	1.1	MATCH	(1.1, 1.0)
$S \rightarrow T$	4.9.2	MATCH	(4.6.2, 0.92), (4.9.2, 1.0), (4.9.14, 0.94)
$S \rightarrow T$	4.9.5	UNMATCHED	na
$S \rightarrow T$	6.2.10	UNMAPPED	(6.2.11, 1.0)
$T \rightarrow S$	6.2.11	ADDITIONAL	na

Passage Ref	AustrianGrid Header	3647 Header
1.1	Overview	Overview
4.6.2	Who may request renewal	Who may request renewal
4.9.2	Who can request revocation	Who can request revocation
4.9.5	Time within which CA must process the revocation request	Time within which CA must process the revocation request
4.9.14	Who can request suspension	Who can request suspension
6.2.10	Cryptographic module rating	Method of destroying private key
6.2.11	n/a	Cryptographic Module Rating

Table 1: Excerpts from a report quantifying the vertical variance of AustrianGrid versus RFC 3647. Row 1 shows that section 1.1 in the Austrian Grid policy exactly matches that of section 1.1 in RFC 3647. However, the mapping from Austrian Grid to RFC 3647 can be more complex. Section headers from the policy under consideration may be ambiguous or not correspond to the accredited policy as shown in rows 2 and 3. Section headers from the accredited policy may be missing in the policy under consideration (as Row 5 seems to indicate for 6.2.11) or relocated. However, looking at Row 4 indicates that section 6.2.11 was moved to section 6.2.10 in the Austrian Grid policy.

Policy	Version	Time (s)	Reff Misses
AustrianGrid	1.2.0	4	0
DFN-PKI	2.1	2	0
DFN-PKI	2.2	2	0
FBCA	2.11	2	0
IRAN Grid	1.3	5	0
IRAN Grid	2.0	2	0
TACC-MICS	1.1	2	0
TACC-Classic	1.2	5	0
TACC-Root	1.2	2	0
ULAGrid	1.0.0	2	0

Table 2: Evaluation 1 shows how we can parse tables of contents to get an inventory of policy sections. For each of the policies, we parse without missing any sections. This indicates that our section inventories accurately reflect the table of contents (TOC).

TOC	$ TOC : TOC + : RFC $	TOC \rightarrow RFC				RFC \rightarrow TOC						
		Mapped	Unmapped	$ TOC $	$ TOC + $	Mapped	Unmapped	$ RFC $				
AustrianGrid	267:267:270	260	260	7	7	267	267	260	260	10	10	270
DFN-PKI-2.1	37:80:270	35	78	2	2	37	80	35	78	235	192	270
DFN-PKI-2.2	79:203:270	75	200	4	3	79	203	75	200	195	70	270
FBCA_CP	281:281:270	242	245	39	36	281	281	242	245	28	25	270
IRAN-GRID-1.3	156:156:193	98	110	58	46	156	156	98	110	95	83	193
IRAN-GRID-2.0	273:273:270	264	264	9	9	273	273	264	264	6	6	270
TACC-MICS_1.1	151:191:270	149	190	2	1	151	191	149	190	121	80	270
TACC_Classic1.2	266:270:270	258	264	8	6	266	270	258	264	12	6	270
TACC_Root_1.2	67:67:270	65	65	2	2	67	67	65	65	205	205	270
ULAGrid_1.0.0	271:271:270	268	268	3	3	271	271	268	268	2	2	270

Table 3: Evaluations 2 and 3 show how well we can classify policy sections as mapped or unmapped. The second evaluation only uses sections from a policy’s table of contents (TOC), which the third evaluation uses an enriched list (TOC+). In 44 sections, we generate a report for the Austrian Grid that successfully identifies a mapping for 260 of the 267 sections in that policy. We added section headers from RFC 3647 to the headers parsed from DFN’s version 2.2 table of contents, resulting in mapping 200 rather than 75 sections.

CP or CPS	$ GroundTruth : TOC + $	GroundTruth- > TOC+			TOC+- > GroundTruth		
		Mapped	Unmapped	$ GroundTruth $	Mapped	Unmapped	$ TOC + $
AustrianGrid	267:267	265	2	267	265	2	267
DFN-PKI-2.1	80:80	79	1	80	79	1	80
DFN-PKI-2.2	207:203	201	6	207	201	2	203
FBCA_CP	309:281	275	34	309	275	6	281
IRAN-GRID-1.3	157:156	145	12	157	145	11	156
IRAN-GRID-2.0	273:273	270	3	273	270	3	273
TACC-MICS_1.1	192:191	188	4	192	188	3	191
TACC_Classic1.2	270:270	267	3	270	267	3	270
TACC_Root_1.2	68:68	67	1	68	67	1	68
ULAGrid_1.0.0	271:271	270	1	271	270	1	271

Table 4: Evaluation 4 shows how well our method in Evaluation 3 approximates actual policy structure. Looking at TACC Root’s CP, we see that only 1 additional provision was identified by manual cataloging rather than automatic extraction. Similarly, only 4 more provisions were identified in DFN-PKI v.2.2. In general our approximation is quite good except for the FBCA CP in which 28, non-standard provisions with citation-depth 4 were identified (e.g. 1.6.2.1).

CP or CPS	$ TOC + / GroundTruth $	$ TOC + Mapped / GroundTruth $
AustrianGrid	100%	99.3%
DFN-PKI-2.1	100%	98.8%
DFN-PKI-2.2	98.1%	97.1%
FBCA_CP	90.9%	89.0%
IRAN-GRID-1.3	99.4%	92.4%
IRAN-GRID-2.0	100%	98.9%
TACC-MICS_1.1	99.5%	97.9%
TACC_Classic1.2	100%	98.9%
TACC_Root_1.2	100%	98.5%
ULAGrid_1.0.0	100%	99.6%

Table 5: Using the results in Table 4, we are able to see that our method in Evaluation 3 was able to identify between 90.9% and 100% of all *actual* provisions. Furthermore, we were able to map the $|TOC + |$ headers to between 89.0% and 99.6% of all *GroundTruth* headers.