2006

An ontology-based distributed whiteboard to determine legal responses to online cyber attacks

Peng, Leisheng

Monterey, California: Naval Postgraduate School.
An ontology-based distributed whiteboard to determine legal responses to online cyber attacks

Leisheng Peng and Duminda Wijesekera
Department of Information and Software Engineering, George Mason University, Fairfax, Virginia, USA

Thomas C. Wingfield
The Potomac Institute for Policy Studies, Arlington, Virginia, USA, and

James B. Michael
Department of Computer Science, Naval Postgraduate School, Monterey, California, USA

Abstract

Purpose – This paper aims to assist investigators and attorneys addressing the legal aspects of cyber incidents, and allow them to determine the legality of a response to cyber attacks by using the Worldwide web securely.

Design/methodology/approach – Develop a decision support legal whiteboard that graphically constructs legal arguments as a decision tree. The tree is constructed using a tree of questions and appending legal documents to substantiate the answers that are known to hold in anticipated legal challenges.

Findings – The tool allows participating group of attorneys to meet in cyberspace in real time and construct a legal argument graphically by using a decision tree. They can construct sub-parts of the tree from their own legal domains. Because diverse legal domains use different nomenclatures, this tool provides the user the capability to index and search legal documents using a complex international legal ontology that goes beyond the traditional LexisNexis-like legal databases. This ontology itself can be created using the tool from distributed locations.

Originality/value – This tool has been fine-tuned through numerous interviews with attorneys teaching and practicing in the area of cyber crime, cyber espionage, and military operations in cyberspace. It can be used to guide forensic experts and law enforcement personnel during their active responses and off-line examinations.

Keywords Systems theory, Law, Decision support systems, Worldwide web

Paper type Research paper

1. Introduction

When a malefactor intrudes into a computer system, the owner of that system – whether a private individual protecting personal property, a corporation securing its assets, or a government defending its interests – needs to know something about the malefactor in order to develop a lawful and effective response to the intrusion. Cyber intrusions may be characterized in one or more of three legal regimes: law enforcement, intelligence collection, and military operations. Furthermore, intrusions can occur across a number of jurisdictional boundaries, building complex conflict-of-laws questions into such attacks. Applying a one-size-fits-all response, such as always terminating all interaction with the intruder, or always responding in kind, can be an
ineffective or worse, illegal, response. For instance, terminating interaction with an intruder could prevent the seizure of evidence for criminal prosecution, collection of information for counterintelligence purposes, or counter-targeting for a military response (Michael, 2002). By responding in kind, the defender may violate domestic or international law or, in the case of a government actor, inadvertently escalate to the level of a use of force or even an armed attack. Furthermore, an inappropriately calibrated response may contravene the customary rules of war accepted as authoritative law by the USA – distinction, necessity, proportionality, and chivalry.

The general problem we address in this paper is that of providing defenders with sufficient information to make informed decisions when formulating responses to the actions of intruders. Specifically, we describe a tool that serves as an automated aid for determining the legal regime under which a cyber intrusion can be categorized, with documentation to support the building of a legal brief. Our tool is built on the premise that owners of affected systems want to defend their computer systems without violating domestic or international law.

Both the frequency and intensity of attacks in cyberspace can be high, affording little time for research and thoughtful consideration before the cyber intrusion (whether a crime, intelligence operation, or an attack) is over. Similarly, what may initially appear to be a minor intrusion or misuse of a computer system may ultimately result in damage to or destruction of property, or even human injury or loss of life. In either case, the defender must be prepared to respond to such attacks with operational plans and mechanisms for real-time information collection already in place; that is, the defender needs to tighten his Observe-Orient-Decide-Act (OODA) loop in order to gain a competitive advantage over the intruder (Michael, 2002).

Legal preparation is an essential element in this equation. Against opponents who disregard any laws which are not immediately and effectively punitive, the default response of inadequately counseled operators is to forego otherwise lawful and effective defensive strategies. In other words, the vast legal gray area that exists today operates in favor of the intruder – a form of asymmetry between the attacker and the defender. A clearer, timelier picture of the operational legalities would provide the defender with more, rather than fewer, intrusion-response options. At this stage in our research, several caveats are in order. First, the present tool is illustrative of the concept, and is not yet intended to be employed operationally. The questions and answers have an artificially academic clarity, which derives from top-down reasoning from broad questions to narrow questions. Second, the decision-tree format no longer defines the state-of-the-art in expert systems, but it does do the following: it presents the core concept clearly; provides a framework with legal ontology that clarifies transparent assembly of resources that support legal analyses; enables the use of a distributed whiteboard for multi-level builders working on one decision tree concurrently without conflicting with each other; and lays a foundation for more elaborate logical structures (such as totality-of-the-circumstances analyses for future operational employment). Third, the inevitable anomalies which will arise in its development (i.e. requiring an early legal determination of whether or not the intruder is a US person will almost certainly conflict with the operational reality of discovering key facts late in the game) serve to highlight conflicts and lacunae in the law. The degree to which the most operationally useful flow of legal questions fails to meet real-world requirements is the degree to which the law or technology must change.
Fourth, and finally, this tool will be developed in alignment with international law, but numerous questions (especially in the law enforcement and intelligence collection realms) will never rise to the level of state versus state legal determinations. Where national and international law appear to conflict, that tension will be made explicit and thus clarified for resolution.

The remainder of this paper is organized as follows: Section 2 describes the details of our application requirements. Section 3 describes the software design of our toolkit. Section 4 explains the functionalities of the decision support system through an example. Section 5 describes related work, and section 6 concludes the paper.

2. Legal requirements
As stated, our objective is to enable timely, effective responses to cyber incidents through legal means and the supporting documentation. In doing so, we are guided by the process of an attorney interviewing his client, establishing the legally operative facts of the case. Our larger objective is to make this a primary global requirement for responding to incidents in a timely manner. In order to reason about response alternatives, we first need a model of the domestic and international law governing cyber intrusions, with one set of decisions for computers to execute at high speed, and another set requiring human decision making at considerably lower speed. Our proposal for this model is a customizable decision tree of legally relevant questions. The computer decision tree will be hardwired for independent execution after meeting clearly discernable, objectively verifiable criteria. This layer of reasoning can be undertaken independent of human intervention, and thus can operate at the speed at which computers interact. The human decision tree will obviously operate far more slowly, and will proceed at the speed of human thought. This tree will present alternative paths requiring deliberate reasoning, and will be equipped with pre-selected sources to assist the attorney in deciding each of the gray-area judgments requiring human reflection and creativity. To do this, it will be necessary to assemble a comprehensive selection of legal sources and append them to each decision point. It will be vital, for speed and clarity, to include no more sources at any given decision point than would be required to answer the question at hand.

During in-depth discussions with experienced attorneys to identify and refine requirements for this system, ontology for organizing and retrieving international and domestic legal source documents was identified as both a useful and urgently. Any two countries may have very different expressions of very similar underlying legal principles. Identifying the core concepts and processes becomes the entry point of forming the legal ontology.

Our tool provides a way to categorize all cyber legal source documents through a general abstracted legal ontology. This general ontology can categorize any cyber sources, and store them in the system database for specific decision tree analytical support or more general thematic research. The ontology contains the principal identifiers of legal sources, such as title, catalog ID, author and credits, keywords, abstract, document type, and nationality. All legal documents may be broadly categorized as constitutional, legislative (statutes), executive (regulations), judiciary (cases), and international. These five categories must be further subdivided into primary (the case or statute itself) and secondary (analytic and synthetic commentary, such as law review articles, or briefs on file). These ten categories are sufficient to
contain any legal source needed to address any given question. Furthermore, each source would have to be presented at four levels of abstraction, for the proper balance of speed and depth:

1. **Citation**: A legal footnote.
2. **Précis**: A sentence or paragraph paraphrasing what the source has to say about the question at hand.
3. **Excerpt**: Direct quotes from the source, which are on point.
4. **Document**: The complete law review article, statute, or case.

Beyond these, there are numerous variables, which must be addressed while "meta-tagging" source materials (see Appendix).

This general information would be distilled into a specific research question in three media: a tree map, which provides a complete visualized tree structure with integrated relationships among tree nodes. This would give a very clear logical flow and the architecture of the entire reasoning process; an audit trail, providing a record of each question asked and each answer chosen; and a brief builder, which would augment the audit trail with those portions of the sources selected by the reviewing attorney to support his answer to the question. This would, in effect, be the first draft of a legal brief supporting the selected course of action.

This decision tree, and its supporting sources, may be constructed using an open source methodology, allowing law students, practitioners, and scholars scattered across the world to collaborate on its construction and refinement. With the process architecture (described below) in place, the trees will be available to selected legal academics for analysis and improvement. Designing such a legal analysis tool for a comprehensive tracking system will be of great benefit to the cyber-legal community, as it will require the analysis and distillation of the entire field into the simplest possible framework for implementation.

This system will take the form of a set of predefined sequence of questions when an actor’s behavior indicates he or she may be intruding into, misusing, or attacking a computer system. To simplify the logic employed, in the prototype, each question has only yes and no answers. A deferent question will follow each yes or no answer to continue the analysis. Then attorneys and their clients would follow a complete logical path to reach a transparently reasoned legal conclusion. A third option, don’t know, allows the user to view the legal resources to evaluate the immediate question, and make a principled decision to proceed forward with a yes or no answer. As mentioned earlier, these resources are arrayed under legal ontology in the Appendix with ten categories (constitutional, legislative, executive, judicial, and international, each at a primary and a secondary level), and each source may be accessed at any one of four levels of abstraction (citation, précis, quotation, and full source).

This system will operate on two levels: for users following previously constructed analyses, and for builders, assembling and testing the analyses to be provided to the operational community. Users are attorneys responsible for providing operational legal advice to law enforcement, intelligence community, or military officials. These users follow a decision tree, answering a sequence of questions carefully crafted to identify and record the legally operative facts of the incident. This decision support tool will produce a logical legal analysis, supported by the legal resources selected by the user.
Builders can be authorized academics, practicing attorneys or appropriately trained computer network technicians.

There are three types of builders: normal builder, super builder, and administrator. Normal builders may add and subtract branches from the decision tree and the resources available at each decision point. These changes, however, will not take effect immediately. A change-request ticket will be created when normal builders try to build a new tree or modify an existing tree. Then system will send these change-request tickets to authorized super builders who have the right to approve or reject the proposed modification. After super builders make these decisions, the approved changes will take effect immediately, and approved and disapproved change-request tickets will be closed by the system. Although multiple normal builders may work on a decision tree and propose changes to the same tree branch at the same time, super builders still have full control of the decision trees and are able to avoid any conflicts during creating and maintaining the substance of the decision support tool. The detail data flow is shown in Figure 1. Administrators, who maintain the authorizations of all builders, may not only build or maintain decision tree directly, but also have the privileges to add and delete a builder, or modify the builder’s login name, password, and the builder types (normal, super or admin).

The tree map is a visualized decision tree with all tree branches and tree nodes. Tree branches show the yes or no links between questions, and tree nodes present the questions with supported legal analysis resource. Current online builders may see who else is online, what changes are being made, and the most recent approved decision tree. By clicking the online builders’ names, builders could compare their own tree maps with others tree maps or the approved decision tree maps instantly. All the tree maps, either changed by multiple builders or approved by super builders, will be displayed in the separate windows. So, for one decision tree, all types of builders can view the differences among those multi-version tree maps. The visualization of decision trees will not only avoid unnecessary redundancy or conflict before normal builders create change-request tickets, but also assist super builders in making the best decisions and guiding the decision tree modification to the correct direction.

![Data flow diagram for distributed whiteboard](image)
3. System design

This prototype is designed to be an open-source, web-enabled decision support tool that provides legal reasoning web services. Multiple clients may access the web server (the system) via web browsers, such as Internet Explorer (IE) or Netscape. The communication language between clients and the web server is HTML exposed within Java Server Pages (JSP). A Java engine (Java 2 Software Development Kit, J2SDK) is used to compile the JSP pages to Java class files that stream HyperText Markup Language (HTML) to web clients and communicate with a mySQL database using JConnector technology. Figure 2 shows the system architecture. Compared to client-server applications, our tool takes full advantages from the multi-tier design:

- Clients may remotely and concurrently access the system, sharing one knowledge base.
- The architecture is extensible, because it is built using the Java 2 Enterprise Edition (J2EE) service framework, with quick deployment times and minimal maintenance efforts in mind. Moreover, the system can be extended to use RDF, OWL, RuleML or JESS as needed.
- The system is easily manageable, because some clients are allowed to change the knowledge base while other clients can only access built-in scenarios.

Each client (a.k.a., actor in the lexicon of software engineering) is a builder or user with his or her own separate applications that share one database and file system. The user’s main functionalities are: answering questions, making a decision, viewing (e.g., audit trial, legal brief, or tree map), or searching and displaying pertinent legal documents under the legal ontology. The builder’s main functionalities are: adding/deleting trees/decisions/questions, linking decisions/resources, loading resource under legal ontology (Appendix), viewing current online builders and their tree maps, approving tickets by super builder, and administrating builder’s group and login information. Details follow.

![System Architecture Diagram](image_url)
3.1 Detailed user requirements (for the forensic examiner)
The following are the detailed user requirements:

- Collect legally relevant facts about the case at hand.
- Follow the decision tree, answering yes, no, or don’t know to each question in sequence. The user can go back to prior questions to change answers in order to see the consequences of alternative answers. A yes or no answer proceeds to the next (pre-determined) question. While many possible paths are available, any given sequence of yes or no answers will result in only one result.
- If the answer is don’t know, the system presents legal resources to assist in making a yes or no decision. These resources will vary in number and length depending on the question at hand, but are grouped by category (Constitutional, Legislative, Executive, Judicial, and International), each with a primary and secondary set of materials, and subcategories such as country and language. Each resource is accessible to four levels of detail: citation, précis, excerpt, and source.
- The audit trail function displays the history of navigation with consulted sources in the citation format along with answers provided by the user.
- The brief builder function will do the same, and include all user-selected portions of all consulted sources.
- After a sufficient number of questions have been answered, the system provides a decision with supporting documents. The user can search databases uploaded into the tool under the ten categories.
- The .txt format legal document is categorized by the legal ontology in the Appendix. Users that search a legal document can get a set of related resources searching by any one or multiple categories of keywords, title, author, citation, précis, excerpt, catalog ID, document type, and country.
- The categorized resource can be used as the help resource by users that chose the don’t know option as the answer to a question on the decision tree.

3.2 Detailed builder requirements (to form the legal basis and strategy)
The following are the detailed builder requirements:

- Three types of builders, normal builder, super builder, and administrator, may login and navigate across any of the tool’s web pages, including those for users. For security reasons, builders inactive for thirty minutes are logged out.
- Allow administrator to maintain all types of builders’ login information, including register new builders, change current builders’ login user name and password, and delete builders who are no longer authorized using builder services any more.
- After initiating a new decision tree or selects an existing one, builders could add question-answer (yes/no/don’t know) and link either of yes or no answer to another question; builders could delete a question, a decision and even an entire tree; builders could upload relevant documents and categorize them in support of the don’t know option. The categorization for uploading legal resource is described in the Appendix.
• Only administrator and super builder can make the above changes directly into the database. The normal builders are only able to send change-request tickets to the super builders and wait for the approval from super builders.

• If a ticket is approved by a super builder, the changes will be effective immediately in system database; if a ticket is rejected by a super builder, requested changes will be discarded.

• The system should be able to locate the specific resource by inputting search criteria.

• Builders should be able to see who else is logged in at the same time.

• For one decision tree, builders can compare different tree maps from different builders, and see the linked questions, decisions and supported resources.

4. System functionality through an example
This section describes the functionality of the tool by constructing an example decision tree to determine the answer to the legal question “Which legal paradigm – law enforcement, intelligence collection, or military operations – applies in this case?” as shown in Figure 3. Again, there are three types of builders: normal builder, super builder, and administrator. All builders may access the system to build a decision tree via a web browser after a correct login. Based on the unique login ID, the system will automatically identify the logged in builder’s type and provide different functionalities according to the pre-authorized rights.

As a first step, the builder creates a new tree “Applicable legal regime”. Then, the builder adds three possible decisions to this new tree. The builder then inserts multiple questions and links the appropriate follow-up questions or decisions to each. The

![Figure 3. An example decision tree](image-url)
builder must specify the parent question to which the new question is to be linked; that is, the builder should design the system so that a yes or no answer to a previous question is linked to a new question posed to a user. Because decision trees can be complex, our system is designed to offer the builder flexibility. The builder can input the system’s decisions and questions without having to enter the links when specifying them. After that, the builder can use menu options to link the decisions and questions. This way a complete decision tree can be constructed in multiple ways, as shown in Figure 4 and Figure 5. When the decision tree needs to be modified because of the changes of law or some other reasons, our system provides sufficient administrative functionalities also, like adding, deleting, re-linking.

When normal builders want to make a change for one decision tree or create a new decision tree, they only can send the change-request tickets, which need to be approved by super builders. So super builders have the global view and full control of decision tree while one or more normal builders work on their individual decision trees. The ticketing approving functionality is shown as Figure 6. The system has a source-searching engine as shown in Figure 7 to locate the original legal document. The search criteria could be very simple: search the user’s input by keyword, title, author, citation, précis, or excerpt. Users could also add document type and country into search criteria. The details of the search criteria for this search engine are shown in Figure 7.

Another important type of functionality in the tool is the distributed whiteboard. The system can provide a list of all online builders, wherever they are located. By clicking the hyper linked online user IDs, system will display the different tree maps in separate windows, as shown in Figure 5. For the same decision tree, different builders may have different change-request tickets to modify that tree. The tree maps displayed by the system will show the approved part of the decision tree, with the pending part of

![Figure 4. A decision rendered by a completed tree](image-url)
Figure 5.
Tree map comparison

Figure 6.
Change-request ticket approval
the decision tree highlighted in orange. Builders can compare the changes with the original tree map and with different versions from other builders. By moving the mouse over each decision node on the tree map, the question for each node will appear automatically. By clicking on one of the nodes with supported resources, the tool will list all available resources for that particular question.

Those resources for supporting analysis are categorized by the legal ontology from the Appendix. In this example, the question “Is the intruder a combatant?” has a resource titled as “United States Code Annotated, Title 18”, and the question “Is the intruder a US Person?” has “Cyber Security Enhancement Act of 2002” as its resource. Table I shows the differences between those two resources after being categorized by the legal ontology. Because different legal systems use different ontologies, the categorization is an interoperable abstract super-ontology that specializes to different legal systems. It not only covers the basic features of normal documents (title, author, keywords, nationality, format, etc.), but also is designed to focus on the special characters of law sources from different legal systems (catalog ID, citation, précis, excerpt, type, etc.). So each legal source can be captured in detail, and searched easily (Figure 8).

5. Related work
Although there have been numerous academic attempts to elicit a logical structure from the legal decision-making process, none is in widespread use with practicing attorneys. The proprietary databases of Westlaw and LexisNexis®, searchable by document category and Boolean keyword strings, are the most frequently consulted. Both have an impressive number of cases, briefs, law review articles, and related documents, like in Hafner’s and Katsh’s papers, but neither is intended to provide direct assistance with the formulation or execution of a legal analysis. Furthermore, there are numerous free sites accessible via the internet – most maintained by universities – that have large, searchable databases. Like their commercial analogs, they provide quick and reliable access to the documents themselves, but are not designed to assist in the legal analysis per se. The University of Minnesota’s Human
<table>
<thead>
<tr>
<th>Categorization field</th>
<th>Resource for question “Is the intruder a combatant?”</th>
<th>Resource for question “Is the intruder a US person?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Cyber Security Enhancement Act of 2002</td>
<td>United States code annotated, title 18</td>
</tr>
<tr>
<td>Catalog ID</td>
<td>H.R. 5710</td>
<td>18 USC 121</td>
</tr>
<tr>
<td>Author &amp; Credits</td>
<td>US Congress</td>
<td>US Congress</td>
</tr>
<tr>
<td>Keywords</td>
<td>Cyber security, sentencing, privacy rights, critical infrastructure, emergency disclosure, good faith exception, illegal devices, provider assistance.</td>
<td>Stored communications, electronic communications, transactional records access, voluntary disclosure</td>
</tr>
<tr>
<td>Citation</td>
<td>Cyber Security Enhancement Act of 2002, H.R. 5710</td>
<td>Stored wire and electronic communications and transactional records access</td>
</tr>
<tr>
<td>Type</td>
<td>Legislative</td>
<td>Legislative</td>
</tr>
<tr>
<td>Nationality</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>Last revision date</td>
<td>2002</td>
<td>2001</td>
</tr>
<tr>
<td>Format</td>
<td>Digital</td>
<td>Digital</td>
</tr>
<tr>
<td>Location</td>
<td><a href="http://www.cybercrime.gov/homeland_CSEA.htm">www.cybercrime.gov/homeland_CSEA.htm</a></td>
<td><a href="http://www.cybercrime.gov/ECPA2701_2712.htm">www.cybercrime.gov/ECPA2701_2712.htm</a></td>
</tr>
</tbody>
</table>

Table I.
Examples instances of a legal ontology

Figure 8.
Search criteria for source searching engine
Rights Library is an excellent example, and is the source of UN Charter text provided in one of our examples.

Capturing legal knowledge and enabling legal discourse is both procedurally and substantively challenging because laws and their interpretation change over time. There are several legal reasoning support tools, such as the ones developed by Muntjewerff and Randall, used primarily to hone law students’ analytical skills. Others are geared for methodology or ontology research. Only a few of these tools are comprehensive web-based tools used for general legal reasoning, and therefore are not specific to one area of law. Duguid and Zeleznikow have developed web-based tools to clarify the application of divorce law. In contrast, when fully developed, our tool may be used not only in training law students and cyberspace technicians, but will also provide real-time legal support for responding to cyber intrusions. Being both web-based and open-source increases the usability, extensibility, maintainability, and capacity for stepwise enhancement of the tool.

6. Conclusion
Those responsible for responding to cyber attacks can benefit from using decision support tools to determine what options for response are available within the applicable legal frameworks. A decision-tree-based tool leads investigators and attorneys through a series of questions that assist them in building legal briefs against cyber intruders. The decisions in a tree are to be constructed by attorneys who are well versed in specific areas of the law: they construct a tree of sequentially-ordered questions that guide the user through to an actionable recommendation for response (i.e. the answer presented at a terminal leaf in the tree). In addition, our toolkit stores the relevant information within well-accepted legal categories (see Appendix) at four levels of detail (citation, précis, excerpt, entire document) necessary to build a legal brief.

Other than being used as decision support tool, this system can also be used as training environment for law students to learn and scale the processes of legal decisions. Because the predefined analysis contained in the system is transparent and well-documented, legal analysis and decision support are completed more accurately and efficiently. Furthermore, the whole system is developed under open-source concept, so the potential usefulness of the tool can support open-source development for a legal ontology.

Reference

Further reading

Appendix. Legal source categorization

(1) Title = main label of the statute.

(2) Catalog ID = publisher or provider’s catalog number. It may be an ISBN.

(3) Author and credits = the author denotes the person(s) or organization(s) responsible for the creation of the statute; credits denotes contributors in addition to the author(s). Examples of authors may include an official body such as a legislature or agency who actually passed a law or regulation; while contributor may include a specific lawmaker’s name who played a key role in moving the law or regulation forward.

(4) Keywords = a list of key words.
Abstract = the main points of the statute. System searches can be made on words in the abstract. Four levels of abstraction: citation, précis, excerpt, and source (entire document).

Type = specifies the type in the broadest terms, such as constitutional, legislative (statutes), executive (regulations), judiciary (cases), and international. The type has subdivision of primary or secondary.

Nationality = country

Metadata version = a designation of the legal database version. This is made up of an organization designation and a numeric indicator of the version.

Length = number of characters or bytes.

New or old.

Expiration date = when a statute may no longer be used.

Last revision date = the date on which the statute was last modified.

Format = the technically defined format of the statute, for instance, the digital format. Format may also define a non-digital format, such as a book or videotape.

Location = typically the URL(s) through which the container can be retrieved, either directly or through an index. Other addressing schemes can accomplish this objective.

User rights = what a user can do with a statute –copyright law; permission rules.

About the authors
Leisheng Peng is currently is a PhD candidate in Information Technology at George Mason University, Department of Information and Software Engineering. Mrs Peng also is a full-time employee of ASRC Management Services since 2004, having the role of Senior Computer Programmer. Previous work includes senior staff responsibilities for Beijing Securities, a top securities company in China, and Webmastering for a Washington, DC university. Mrs Peng completed her MS in Computer Science in 2001 at Southeastern University and her BS in Economics in 1997 at The Finance and Banking Institute of China. Leisheng Peng is the corresponding author can be contacted at: lpeng@gmu.edu

Duminda Wijesekera is an Associate Professor in the Department of Information and Software Engineering at George Mason University, Fairfax, Virginia. During various times, he has contributed to research in security, multimedia, networks, systems, avionics and theoretical computer science. These span topics such as applying logical methods to access and dissemination control, securing circuit switched (SS7) and IP-based telecommunication systems, multimedia, security requirements processing during the early phases of the software lifecycle, WWW security, railroad signaling security, SCADA security and engineering Ballistic Missiles. His pre-GMU work has been in quality of service issues in multimedia, avionics control and specifying and verifying concurrent systems using logical methods. He holds courtesy appointments at the Center for Secure Information Systems (CSIS) and the Center for Command, Control and Coordination (C4I) at George Mason University, and the Potomac Institute of Policy Studies in Arlington, VA. He is also the Director of Liaisons at Aeolus Systems, an engineering services provider based in Clearwater, Florida and Nashua, New Hampshire. Prior to joining GMU as an assistant professor in 1999, he was a senior systems engineer at Honeywell Space Systems in Clearwater, Florida. He has been a visiting post-doctoral fellow at the Army High Performance Research Center at the University of Minnesota, and an assistant professor of Mathematics at the University of Wisconsin. Wijesekera received a PhD in Computer Science from the University of Minnesota in 1997 under Jaideep Srivastava and a PhD in Mathematical Logic from Cornell University in 1990 under Anil Nerode.
Thomas C. Wingfield is an international lawyer with a specialty in national security law. He holds a BA in History and Russian Language (summa cum laude) from Georgia State University, a Doctor of Laws (JD) and Master of Laws (LLM, with distinction, International and Comparative Law) from the Georgetown University Law Center, and is pursuing his second doctorate, a Doctor of Juridical Science (SJD, National Security Law) at the Law School of the University of Virginia. Beginning his career as a naval officer, he served as Squadron Intelligence Officer with an F/A-18 strike fighter squadron aboard USS Midway, as a HUMINT Desk Officer within the Office of Naval Intelligence, and as Intelligence Liaison Officer at the Center for Naval Analyses. Following his naval service, Mr Wingfield took a position with ManTech Corporation and advised military and intelligence community clients in the areas of treaty compliance, use of force in cyberspace, and space law. Since August, 2003, he has been a Research Fellow at the Potomac Institute, advising clients within Congress and the Administration on the legal aspects of various national security issues. Mr Wingfield is a Lecturer in Law at Columbus Law School (Catholic University of America) in Washington, DC, and a former Chair of the American Bar Association’s Committee on International Criminal Law. He has lectured widely and written extensively on cyber conflict, tyranny and democracy, and lawful uses of force in the war on terror. He is the author of the legal text, The Law of Information Conflict: National Security Law in Cyberspace.

James B. Michael is an Associate Professor of Computer Science and holds a joint appointment as an Associate Professor of Electrical & Computer Engineering. Bret joined the faculty of the Naval Postgraduate School in May 1998 and was awarded tenure in July 2003. Prior to joining NPS, he was an Associate Research Engineer with the University of California at Berkeley (1994-1998), conducting research on automated vehicle control and safety systems for automated highway systems. He served as a Formal Methods Engineer for Argonne National Laboratory (1992-1993) and a Member of the Research Staff at the Institute for Defense Analyses (1988-1992), in both positions conducting research on what was then known as the Strategic Defense Initiative (a.k.a. “Star Wars”). Bret’s research centers on both building dependable distributed systems and assessing their trustworthiness. Bret is co-authoring two books on this subject: Delivering Dependable Software Systems (John Wiley & Sons, Hoboken, NJ), co-authored with Dr Jeffrey Voas of SAIC, Inc., and Distributed Battle Management Engineering and C3 for Missile Defense (Artech House, London), with his colleague Professor Phillip Pace. Bret was elected in October 1997 to the grade of senior member of the Institute of Electrical and Electronics Engineers (IEEE). He was appointed to the IEEE Reliability Society’s Administrative Committee in 2005 and is working with John Harauz of Jonic Systems Engineering, Inc. to form the IEEE Task Force on System Safety. Bret is also a long-time member of both the Association for Computing Machinery (ACM) and the International Federation for Information Processing (IFIP). He serves on several advisory boards and steering committees for the US Government, including the Steering Committee of the Department of Defense Information Assurance Technology Analysis Center (IATAC). He also serves on the editorial board for the Journal of Information and Management (Elsevier) and the board of advisors for IEEE Software. He received his PhD degree from the George Mason University School of Information Technology and Engineering, Fairfax, Virginia in May 1993.