

EXHIBIT G

Application of: Doyle et al.

Application Num.: 90/007,858

Filed: December 22, 2005

For: Distributed Hypermedia Method for Automatically Invoking External Application Providing Interaction and Display of Embedded Objects Within a Hypermedia Document

Examiner: Pokrzywa, J.R.

Art unit: 3992

Declaration of Edward W. Felten

I, Edward W. Felten, declare as follows:

1. I have been retained by Eolas and the Regents of the University of California to serve as an expert in the field of computer science and Internet software. My Curriculum Vitae, which recites my technical expertise, is attached hereto to as Exhibit A.

I. Qualifications

2. I graduated with Honors from the California Institute of Technology in 1985, with a B.S. degree in Physics. I received an M.S. in Computer Science in 1991, and a Ph.D. in Computer Science in 1993, both from the University of Washington.

3. I am currently a Professor of Computer Science and Public Affairs at Princeton University, where I have taught since 1993. I was originally hired at Princeton as an Assistant Professor of Computer Science, in 1993. I was promoted to Associate Professor of Computer Science in 1999, and to Professor of Computer Science in 2003. I was appointed as Professor of Computer Science and Public Affairs in 2006.

4. I am the author or co-author of numerous publications relating to computer science and Internet software. These publications are listed in my CV.

5. I have been asked to address the arguments presented in the Office Action mailed July 30, 2007 ("the Office Action") in connection with the reexamination of United States Patent No. 5,838,906 ("the '906 Patent") that the claims of the '906 Patent are unpatentable as being "anticipated" by U.S. Patent No. 5,367,621 ("Cohen"), issued to Cohen. For the reasons described in this declaration, I respectfully disagree with the arguments presented in the Office Action and, instead, believe that the claims of the '906 Patent fully meet the requirements for patentability over the cited references, as those patentability arguments have been described to me.

6. To familiarize myself with the issues involved in the rejection of the claims, I have reviewed numerous documents, including the following: the '906 Patent and its file history, the documents cited in the Office Action, and all other documents referenced or cited in this declaration.

7. My discussion is also based on my experience as a computer science researcher and teacher, and as a Web user and network software developer. From this experience, I have gained an independent understanding of how the browser art developed.

II. Response to the Unpatentability Arguments Raised in the Office Action

8. I have been told by patent counsel for Eolas and the Regents that a patent may not be obtained if the patent's subject matter is anticipated by a prior art reference. I have further been told that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. I have further been told that to serve as an anticipation when a reference is silent about the asserted inherent characteristic, such gap in the reference may be filled with recourse to extrinsic evidence, and that such evidence must make it clear that the missing descriptive matter is necessarily present in the thing described in the reference and that it would be so recognized by persons of ordinary skill in the art.

9. In reading the claims of the '906 Patent, I am using the claim construction established by Judge Zagel and upheld by the Federal Circuit in the *Eolas v. Microsoft* litigation.

1. The Level of Skill in the Art

10. My benchmark for what ordinary skill in the art means is a person who is just graduating from a good computer science program at a college or a university – not a star student but just an average student – or a person who has gained an equivalent level of knowledge through experience in the industry. This person knows how to do things in conventional ways but does not exhibit an unusual level of innovative thinking.

11. In 1994, those of ordinary skill in the art were just becoming familiar with the Web and Web browsers. One of ordinary skill would have had a general idea of how the Mosaic browser worked, and would have been familiar with hyperlinks, forms, and helper applications.

2. The '906 Patent

12. The claims of the '906 Patent describe a technology that allows web page authors to include, within the boundaries of a web page, interactive objects. This is done (briefly stated) by including in the web page's HTML text an embed text format, that provides information about where to get the object's data, along with information to identify and locate an executable application that will be invoked on the client computer to display the data and to provide interactivity with it, and by providing a web browser that knows how to parse the HTML to extract the embed text format, how to use type information to identify and locate the

executable application, how to invoke the executable application, to execute on the client computer, and how to interface to the executable application so as to allow the user to interact with it within the boundaries of the browser window.

13. My analysis of patentability will focus on Claim 6 of the '906 Patent. Claim 1 is very similar to Claim 6, the only material difference being that Claim 6 covers a computer program product and Claim 1 covers a method for using such a product. Accordingly, all of my analysis of Claim 6 applies to Claim 1 as well. Patentability of the other dependent claims discussed in the Office Action follows from the patentability of the underlying independent Claims 1 and 6.

3. Prior Art Browsers, Including Mosaic

14. The Office Action cites the applicants' admitted prior art. I have reviewed all prior art references referenced in the '906 Patent's file history. The Office Action's discussion of this prior art focuses on the Mosaic browser, which was the most advanced prior art browser.

15. In 1994, the Web was young, and browsers were a relatively new technology. Browsers offered only a very limited form of interactivity. A page could contain hyperlinks, on which the user could click to view another page. A page could be a form to be filled out by the user, with a "submit" button which, when clicked, caused the user to see another page.

16. Another technology, known as "helper applications," was implemented in the Mosaic browser. This technology allowed the browser to link to an external program, in cases where the browser encountered a file whose format the browser did not understand. For example, if the user clicked on a hyperlink that pointed to a file in .mpeg format (i.e., a movie in MPEG format), then the browser would launch an external MPEG-viewer program and pass the .mpeg file to that program. The result would be that the MPEG program ran, in a separate window from the browser.

17. Helper applications allowed the browser to link to an external program, but that program could not provide interactivity within the browser window. The helper application was just an external program that ran on the same computer, in a separate window.

18. None of these methods allowed a Web page author to place fully interactive objects within the confines of a Web page's display.

19. These methods are all implemented in today's browsers, and they are all in use on the Web today.

A. '906 Claim Elements Lacking in Mosaic

20. Mosaic lacked at least three elements required by the claims of the '906 Patent. Among the required elements lacking in Mosaic were an embed text format, automatic invocation, and display of the object within the browser window.

i. Mosaic Lacked the Required Embed Text Format

21. Mosaic lacked the required embed text format of the '906 claims. Instead, Mosaic used an ordinary hyperlink to link to any data that was to be displayed with a helper application.

22. Claim 6 requires that "said first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object..." ('906 Patent at 18:14-18)

23. Hyperlinks in Mosaic were specified using an "A" (short for "Anchor") tag. For example, the HTML element

    link

    would cause the text "link" to be displayed, typically in underlined blue type. If the user clicked on the underlined word "link", the browser would follow the hyperlink and navigate to the URL "http://example.com/page.html".

24. Similarly, the HTML element

    video

    would cause the text "video" to be displayed, in underlined blue type. If the user clicked on the underlined word "photo", the browser would download the file at the URL "http://example.com/video.mpg" and launch a helper application to display it.

25. In the claimed '906 system, the browser instead used a special tag, the "embed text format", to specify that an embedded object should be included. Mosaic lacked the embed text format. The use of an embed text format was a significant improvement over the prior art Mosaic browser, as it allowed the browser to recognize immediately that an embedded object was present and special processing was needed.

26. The Examiner states that Mosaic lacks the embed text format: "the NCSA Mosaic references ... are not seen to teach the combination of limitations requiring executing a browser application at said client workstation that parses a first distributed hypermedia document, having an **embed text format included in the hypermedia document that specifies the location of at least a portion of an object external to the first distributed hypermedia document**..." (Office Action at paragraph 14, emphasis in original)

ii. Mosaic Lacked the Required Automatic Invocation

27. Mosaic further lacked the required automatic invocation of the '906 claims.

28. Claim 6 includes the limitation "wherein said embed text format is parsed by said browser to automatically invoke said executable application" ('906 Patent at 18:23-25), which requires that the executable application be invoked automatically, that is, without requiring any action such as a mouse click from the user.

29. In Mosaic, as stated above, a helper application was launched via an ordinary hyperlink. All hyperlinks required a mouse click by the user to be activated. To launch a helper application, the user had to make such a mouse click. Therefore, the launching of helper applications was not automatic.

30. Automatic invocation was an important improvement in the claimed '906 technology over the prior art Mosaic browser. Automatic invocation allowed the object to appear immediately when the user visited the enclosing web page, thus helping to make the object appear to be an integral part of the web page.

iii. Mosaic Lacked the Required Display within the Browser Window

31. Mosaic further lacked the required '906 claim element of displaying the object within the browser window.

32. Claim 6 of the '906 Patent includes the limitation "to display said object and enable interactive processing of said object within a display area created at said first location within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window." ('906 Patent at 18:26-30) This requires the object to be displayed, and interactive processing of the object to be enabled, within an area in the browser window.

33. Mosaic helper applications displayed in a separate window. When a helper application was launched, a new window was created, and the helper application used this new window exclusively. The new window used by the helper application was entirely separate from any window used or controlled by the browser.

34. The ability of the claimed '906 technology to display and enable interactive processing within the browser window was a significant advance over the prior art Mosaic browser. Enabling display and interactivity within the browser window allowed the object to appear, seamlessly, as an integral part of the web page's display. This was a more natural model for users, allowing the object to be placed in the context of any descriptive text appearing on the web page.

4. The Cohen Reference

35. Cohen describes a system that allows the author of a softcopy book to specify specific multimedia presentations accompanying the text of the book.

B. Cohen Does Not Anticipate the '906 Claims

36. The Examiner argues that the claims 1-3 and 6-8 of the '906 Patent are anticipated by Cohen. I respectfully disagree with this conclusion, for the reasons given below.

37. As discussed above, my analysis will focus on Claim 6 of the '906 Patent. My analysis of Claim 6 also applies to Claim 1. Non-anticipation of the dependent Claims 2, 3, 7, and 8 follows from non-anticipation of the underlying independent Claims 1 and 6.

38. Several elements of Claim 6 are absent in Cohen. For example, Cohen lacks the elements of interactive processing, an embed text format, display of the object within a browser window, and the use of type information by a browser to identify and locate an executable application.

C. Cohen Lacks the Required Interactive Processing

39. Claim 6 of the '906 Patent requires that the browser "automatically invoke said executable application to execute on said client workstation in order to display said object and **enable interactive processing of said object** ..." ('906 Patent at 18:24-27, emphasis added) Cohen lacks the element of interactive processing.

40. Claim 6 requires that the system not only "display said object" but also "enable interactive processing" of the object. To display the object is to make it visible, e.g. by sending it to an output device. Interactive processing means that the user, by using the mouse or keyboard or similar input device, can change the structure or presentation of the object. The interactive processing must occur within the "display area" where the object is displayed.

41. Cohen does not expressly disclose interactive processing of the object. Cohen repeatedly discusses displaying the object, but nowhere is this display coupled to interactive processing of the object. See, e.g. Cohen at 2:44-47 ("transferring ... to the multimedia output device, the multimedia data from the multimedia object"), 2:59-60 ("to support the presentation of the multimedia object"), 8:31-35 (describing how an animation object is "displayed"), 8:46-56 ("displayed graphics"), 9:46-10:4 ("to transfer the audio data from the audio object to the stereo high fi to produce the audio presentation .... to output audio data from the audio object stored thereon to the stereo high fi for the audio presentation ...."), 12:4-23 ("playing of the video information from the compact disk player ... and the presentation of the resulting motion picture on the display 208 at the workstation"), 13:7-9 ("the animation for the ALT_OBJECT is displayed"), 13:21-51 ("Audio data form [sic] the object is output to the audio support."), 13:52-14:10 ("The data in the object 110 is output to the appropriate graphic support software and the graphics is generated ... as the graphics 190' which is displayed on the display 208.") Nowhere does Cohen discuss **interactive processing** of the object within a browser window.

42. The Examiner does not point to any support in Cohen for the element of interactive processing. The Examiner discusses, as a single unit, a longer portion of the claim (Office Action at p. 32), and the Examiner cites support in Cohen for some aspects of this longer portion of the claim. But none of the material cited by the Examiner supports the notion that Cohen discloses **interactive processing** of an object within a browser-controlled window.

43. Nor is interactive processing inherent in Cohen. Displaying an object does not necessarily require that the user can interact with the object.

44. The design of Cohen was incompatible with interactive processing. In Cohen, when the main book reader program invokes an external program, it waits for the external program to finish before resuming the main book reader program. While

the external program is running, the main book reader program is frozen and cannot accept input. This is evident from Fig. 6 of Cohen and the accompanying description in (e.g.) Columns 10 and 11. For example:

> After the completion of the I/O handler program whose execution was started as a result of step 426 of the flow diagram of FIG. 6, the flow proceeds back to step 420 to wait for another mouse pointer to activate a link tag or alternatively to wait until a new page is requested by the user.

(Cohen at 11:22-27) Notably, it is only after the completion of the external program that the main book reader program can accept input from the user. If the external program had tried to provide interactive processing, this would have required the external program to continue running (waiting for user interaction) which would have frozen the main book reader program, giving the user no way to switch to a different page of the e-book. The role of the external program in Cohen is to display the object and then stop running.

45. By contrast, in the invention defined in Claim 6 of the '906 Patent, the browser and the external program can both be active and accepting input at the same time, so the user has the option of seamlessly interacting with the object or navigating the browser.

46. One of the main achievements of the '906 inventors was to provide interaction with an object, seamlessly within a Web page. A system like Cohen, which can display objects but cannot provide interactive processing, is missing an important element of the invention defined in Claim 6 of the '906 Patent.

D. Cohen Lacks the Required Embed Text Format

47. Claim 6 of the '906 Patent requires that "said first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, ..., and wherein said embed text format is parsed by said browser to automatically invoke said executable application .. in order to display said object and enable interactive processing of said object within a display area created at said first location within the portion of the first distributed hypermedia document ..." ('906 Patent at 18:14-29)

48. The use of an embed text format is an important element of the invention defined in Claim 6 of the '906 Patent. One drawback of many prior art browsers, such as Mosaic, is that they lacked an embed text format. For example, the examiner states that "the NCSA Mosaic references ... are not seen to teach the combination of limitations requiring executing a browser application at said client workstation that parses a first distributed hypermedia document, having an **embed text format included in the hypermedia document that specifies the location of at least a portion of an object external to the first distributed hypermedia document**..." (Office Action at paragraph 14, emphasis in original)

49. Cohen does not disclose the use of an embed text format.

50. The Examiner states that the "link description tags LDESC" of Cohen are the embed text format (Office Action at p. 31). For the reasons described below, I respectfully disagree with this conclusion.

51. The LDESC tags cannot be the embed text format, because they do not satisfy the required claim element "wherein said first distributed hypermedia document includes an **embed text format, located at a first location** in said first distributed hypermedia document ... to display said object and enable interactive processing of said object **within a display area created at said first location** ..." ('906 Patent at 18:14-28, emphasis added) This claim element requires that the embedded object be displayed at a location in the distributed hypermedia document (e.g., the Web page) that corresponds to the location of the embed text format within the document.

52. The LDESC tag does not appear in the document at the required location. Instead, the LDESC (link description) tag appears in the document file's prologue. "The BookMaster tags are improved upon, in accordance with the invention, to provide a new multimedia link description tag LDESC **in the prologue of the document** ..." (Cohen at 5:8-11, emphasis added) The figures in Cohen also depict the LDESC tags as being in the file's prologue rather than in the book text. For example, Figure 1 depicts the structure of the document file. The "link description tags 102" (also depicted in Figure 1a) are distinct from the "book text with tags 104" (which is also depicted in Figure 1b). (See also Figs. 4, 8a, and 8b.)

53. The fact that the LDESC tag does not appear at a location in the book text is one reason why the LDESC tag cannot be the embed text format of the '906 claims.

54. The "link tag :L" of Cohen does appear in the book text, but it cannot be the embed text format either, because (e.g.) it lacks the required claim element of an "embed text format ... that specifies the location of at least a portion of an object external to the first distributed hypermedia document ..." ('906 Patent at 18:15-18). The link tag of Cohen does not specify the location of an object, nor does it specify the location of anything that is external to the first distributed hypermedia document. This is one reason why the "link tag :L" of Cohen cannot be the embed text format.

55. Cohen's design strategy, of having a small, simple link tag that refers to a larger, more detailed link description in the document prologue, makes sense given the problem that Cohen was trying to solve. Cohen was designed for use with electronic books. These books, unlike Web pages, are large, multi-page files that often repeat graphic elements on different pages. By separating the link tag and link description, Cohen allowed an element to be repeated without having to repeat the full link description each time. Instead, there could be a single link description in the document prologue, and one small link tag at each place in the document where the object was to be used. The claimed '906 design, by contrast, is better suited for use on the Web where individual pages are provided separately. For at least this reason, the use of a single tag is not expressly found or inherently described in Cohen.

56. Another reason that Cohen and Mosaic do not anticipate Claim 6 is that neither Cohen nor Mosaic had an embed text format.

E. Cohen Lacks the Required Display within the Browser Window

57. Claim 6 of the '906 Patent requires that the display of the object take place "within a display area ... within said first browser-controlled window" ('906 Patent at 27-30). Cohen lacks the required element of displaying the object within the browser-controlled window.

58. Cohen discloses displaying the object on a screen, but there is no disclosure of the display happening within an area of the browser-controlled window.

59. The Examiner does not point to any support in Cohen for the element of displaying within the browser-controlled window. The Examiner discusses, as a single unit, a longer portion of the claim (Office Action at p. 32), and the Examiner cites support in Cohen for some aspects of this longer portion of the claim. But none of the material cited by the Examiner supports the notion that Cohen discloses displaying of the object within a display area in the browser-controlled window.

60. Displaying within the browser-controlled window is not inherent in Cohen either. In Cohen, displaying does not **necessarily** take place within the browser-controlled window. The disclosure in Cohen is consistent with the display taking place in a separate window, as with Mosaic helper applications, or on a separate device.

61. One reason that Claim 6 is not anticipated by Cohen and Mosaic is that both Cohen and Mosaic lack the require element of displaying the object within a display area in the browser-controlled window.

F. Cohen Lacks the Required Use of Type Information by the Browser to Identify and Locate an Executable Application

62. Claim 6 of the '906 Patent requires, "wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document..." ('906 Patent at 18:19-22). Cohen lacks this element.

63. The Examiner's only discussion of this element of Claim 6 in relation to Cohen is as follows:

    > See Fig. 1a, whereby the external executable application is specified as "DATA = 'graph.exe \GOCA FORMAT C'", seen in Fig. 1a, wherein the graph.exe program is external to the hypermedia document

    (Office Action at pp. 31-32) In this discussion, the Examiner is apparently reading the "DATA = ..." line as being the type information, and graph.exe as being the executable application. However, the Examiner does not point to any disclosure of a browser utilizing the type information to identify and locate the executable application. Nor does Fig. 1a of Cohen depict any steps performed by the browser.

64. In Cohen, the book reader (which the Examiner equates to a browser) does not utilize the "type information" to identify and locate anything. All the book reader does with this information is to pass it on, unexamined, to the operating system, which invokes the application. The book reader does not have any kind of algorithm or procedure that it follows to identify and locate an application to be used.

65. The identify and locate step, and the fact that that step is done by the browser, is an important aspect of the invention defined in Claim 6 of the '906 Patent. For example, this step provides an important security protection. Users often want to display distributed hypermedia documents that come from untrusted sources, such as Web pages that come from arbitrary sites. If the author of such a site can cause an executable application of his choice to be invoked on the user's system, then the site author can use that application to gain access to the user's private files or modify the state of the user's computer, for example to install spyware or a virus.

66. Having the browser – a program trusted by the user – identify and locate the executable application lets the browser protect the user from this danger. A properly written browser will only allow trusted applications to be run, thereby protecting the user against security problems. A hostile site author cannot run a malicious application on the user's computer, because it is the browser, not the site author, that is identifying and locating the application that will be run. References in which the browser does not utilize type information to identify and locate the executable application lack this protection.

67. The importance of this claim element was reinforced by the *Eolas v. Microsoft* litigation, in which the question of which references had this claim element was an important one.

III. Conclusion

68. For the reasons given above, I conclude that the rejection of claims 1 and 6 as being unpatentable is incorrect. The claims of the '906 patent were not anticipated by the references cited in the Office Action.

I declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under 18 U.S.C. Section 1001, and that such willful false statements may jeopardize the validity of the patent.

Dated: September 27, 2007

Edward W. Felten

Edward W. Felten

Edward W. Felten

Dept. of Computer Science
Princeton University
35 Olden Street
Princeton NJ 08540
(609) 258-5906
(609) 258-1771 fax
felten@cs.princeton.edu

Education

Ph.D. in Computer Science and Engineering, University of Washington, 1993.
Dissertation title: "Protocol Compilation: High-Performance Communication for Parallel Programs." Advisors: Edward D. Lazowska and John Zahorjan.
M.S. in Computer Science and Engineering, University of Washington, 1991.
B.S. in Physics, with Honors, California Institute of Technology, 1985.

Employment

Professor of Computer Science, Princeton University, 2003-present.
Associate Professor of Computer Science, Princeton University, 1999-2003.
Assistant Professor of Computer Science, Princeton University, 1993-99.
Senior Computing Analyst, Caltech Concurrent Computing Project, California Institute of Technology, 1986-1989.

Director, Secure Internet Programming Laboratory, Dept. of Computer Science, Princeton University, 1996-present.

U.S. Dept. of Justice, Antitrust Division: consulting and testimony in Microsoft antitrust case, 1998-2002.
Robins Kaplan, Miller & Ciresi. Consulting and testimony in patent lawsuit, 1998-2003.
Keker & Van Nest. Consulting in intellectual property / free speech lawsuit, 2002.
Electronic Frontier Foundation. Consulting in intellectual property / free speech lawsuits, 2001-present.
Certus Ltd.: consultant in product design and analysis, 2000-2002.
Cigital Inc.: Technical Advisory Board member, 2000-present.
Cloakware Ltd.: Technical Advisory Board member, 2000-present.
Propel.com: Technical Advisory Board member, 2000-2002.
NetCertainty.com: Technical Advisory Board member, 1999-2002.
FullComm LLC: Scientific Advisory Board member, 1999-2001.
Sun Microsystems: Java Security Advisory Board member, 1997-present.
Finjan Software: Technical Advisory Board member, 1997-2002.

EXHIBIT A

International Creative Technologies: consultant in product design and analysis, 1997-98.
Bell Communications Research: consultant in computer security research, 1996-97.

Honors and Awards

Scientific America 50 Award, 2003.
Alfred P. Sloan Fellowship, 1997.
Emerson Electric, E. Lawrence Keyes Faculty Advancement Award, Princeton University School of Engineering, 1996.
NSF National Young Investigator award, 1994.
Outstanding Paper award, 1997 Symposium on Operating Systems Principles.
Best Paper award, 1995 ACM SIGMETRICS Conference.
AT&T Ph.D. Fellowship, 1991-93.
Mercury Seven Foundation Fellowship, 1991-93.

Research Interests

Computer security, especially relating to consumer products. Technology law and policy. Internet software. Operating systems. Interaction of security with programming languages and operating systems. Distributed computing. Parallel computing architecture and software.

Professional Service

Professional Societies and Advisory Groups

ACM Advisory Committee on Security and Privacy, 2002-2003.
DARPA Information Science and Technology (ISAT) advisory group, 2002-present.
Co-chair, ISAT study committee on "Reconciling Security with Privacy," 2001-2002.
National Academy study committee on Foundations of Computer Science, 2001-present.

Program Committees

USENIX General Conference, 2004.
Workshop on Foundations of Computer Security, 2003.
ACM Workshop on Digital Rights Management, 2001.
ACM Conference on Computer and Communications Security, 2001.
ACM Conference on Electronic Commerce, 2001.
Workshop on Security and Privacy in Digital Rights Management, 2001.
Internet Society Symposium on Network and Distributed System Security, 2001.
IEEE Symposium on Security and Privacy, 2000.
USENIX Technical Conference, 2000.
USENIX Windows Systems Conference, 2000.
Internet Society Symposium on Network and Distributed System Security, 2000.

IEEE Symposium on Security and Privacy, 1998.
ACM Conference on Computer and Communications Security, 1998.
USENIX Security Symposium, 1998.
USENIX Technical Conference, 1998.
Symposium on Operating Systems Design and Implementation, 1996.

Corporate Advisory Boards

Sun Microsystems, Java Security Advisory Council.
Cigital Inc.: Technical Advisory Board.
Cloakware Ltd.: Technical Advisory Board.
Propel.com: Technical Advisory Board.
Finjan Software: Technical Advisory Board.
Netcertainty: Technical Advisory Board.
FullComm LLC: Scientific Advisory Board.

University and Departmental Service

Faculty Advisory Committee on Policy, 2002-present.
Council of the Princeton University Community, 2002-present (Executive Committee)
Faculty Advisory Committee on Athletics, 1998-2000.
Computer Science Academic Advisor, B.S.E. program, class of 1998 (approx. 25 students)
Faculty-Student Committee on Discipline, 1996-98.
Faculty-Student Committee on Discipline, Subcommittee on Sexual Assault and Harrassment, 1996-98.

Students Advised

Ph.D. Advisees:

Minwen Ji (Ph.D. 2001). Dissertation: Data Distribution for Dynamic Web Content. Researcher at Compaq Systems Research Center.
Dirk Balfanz (Ph.D. 2000). Dissertation: Access Control for Ad Hoc Collaboration. Researcher at Xerox Palo Alto Research Center.
Dan S. Wallach (Ph.D. 1998). Dissertation: A New Approach to Mobile Code Security. Assistant Professor of Computer Science, Rice University.
Robert A. Shillner (Ph.D. expected 2004). Tentative dissertation title: Improving Distributed File Systems using a Shared Logical Disk. Technical staff member at Google.
Michael Schneider (Ph.D. expected 2003). Dissertation topic: Network Defenses against Denial of Service Attacks.

Significant Advisory Role:

Drew Dean (Ph.D. 1998). Advisor: Andrew Appel. Researcher at SRI International.
Stefanos Damianakis, Ph.D. 1998. Advisor: Kai Li. President, Netrics, Inc.
Pei Cao, Ph.D. 1996. Advisor: Kai Li. Assistant Professor of Computer Sciences, University of Wisconsin. On leave at Cisco Systems.

Lujo Bauer, Ph.D. 2003. Advisor: Andrew Appel. Postdoctoral researcher at Carnegie-Mellon University.

Publications

Books and Book Chapters

[1] Freedom to Tinker. Edward W. Felten. Publication expected, 2004.

[2] Securing Java: Getting Down to Business with Mobile Code. Gary McGraw and Edward W. Felten. John Wiley and Sons, New York 1999.

[3] Java Security: Web Browsers and Beyond. Drew Dean, Edward W. Felten, Dan S. Wallach, and Dirk Balfanz. In "Internet Besieged: Countering Cyberspace Scofflaws," Dorothy E. Denning and Peter J. Denning, eds. ACM Press, New York, 1997.

[4] Java Security: Hostile Applets, Holes and Antidotes. Gary McGraw and Edward Felten. John Wiley and Sons, New York, 1996.

[5] Dynamic Tree Searching. Steve W. Otto and Edward W. Felten. In "High Performance Computing", Gary W. Sabot, ed., Addison Wesley, 1995.

Journal Articles

[6] Mechanisms for Secure Modular Programming in Java. Software – Practice and Experience, 33:461-480, 2003.

[7] The Digital Millennium Copyright Act and its Legacy: A View from the Trenches. Illinois Journal of Law, Technology and Policy, Fall 2002.

[8] DRM and Fair Use: A Skeptical View. Edward W. Felten. Communications of the ACM. April, 2003.

[9] The Security Architecture Formerly Known as Stack Inspection: A Security Mechanism for Language-based Systems. Dan S. Wallach, Edward W. Felten, and Andrew W. Appel. ACM Transactions on Software Engineering and Methodology, 9:4, October 2000.

[10] Statically Scanning Java Code: Finding Security Vulnerabilities. John Viega, Tom Mutdosch, Gary McGraw, and Edward W. Felten. IEEE Software, 17(5), Sept./Oct. 2000.

[11] Client-Server Computing on the SHRIMP Multicomputer. Stefanos N. Damianakis, Angelos Bilas, Cezary Dubnicki, and Edward W. Felten. IEEE Micro 17(1):8-18, February 1997.

[12] Fast RPC on the SHRIMP Virtual Memory Mapped Network Interface. Angelos Bilas and Edward W. Felten. IEEE Transactions on Parallel and Distributed Computing, February 1997.

[13] Implementation and Performance of Integrated Application-Controlled File Caching, Prefetching and Disk Scheduling. Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. ACM Transactions on Computer Systems, Nov 1996.

[14] Virtual Memory Mapped Network Interface Designs. Matthias A. Blumrich, Cezary Dubnicki, Edward W. Felten, Kai Li, and Malena Mesarina. IEEE Micro, 15(1):21-28, February 1995.

Symposium Articles

[15] Receiver Anonymity via Incomparable Public Keys. Brent R. Waters and Edward W. Felten. ACM Conference on Computer and Communications Security. November 2003.

[16] Attacking an Obfuscated Cipher by Injecting Faults. Matthias Jacob, Dan Boneh, and Edward W. Felten. ACM Workshop on Digital Rights Management, November 2002.

[17] A General and Flexible Access-Control System for the Web. Lujo Bauer, Michael A. Schneider, and Edward W. Felten. 11th USENIX Security Symposium, August 2002.

[18] Informed Consent in the Mozilla Browser: Implementing Value-Sensitive Design. Batya Friedman, Daniel C. Howe, and Edward W. Felten. Hawaii International Conference on System Sciences, January 2002. (Best Paper award, organizational systems track.)

[19] Reading Between the Lines: Lessons from the SDMI Challenge. Scott A. Craver, John P. McGregor, Min Wu, Bede Liu, Adam Stubblefield, Ben Swartzlander, Dan S. Wallach, Drew Dean, and Edward W. Felten. USENIX Security Symposium, August 2001.

[20] Cookies and Web Browser Design: Toward Realizing Informed Consent Online. Lynette I. Millett, Batya Friedman, and Edward W. Felten. Proc. of CHI 2001 Conference on Human Factors in Computing Systems, April 2001.

[21] Timing Attacks on Web Privacy. Edward W. Felten and Michael A. Schneider. Proc. of 7th ACM Conference on Computer and Communications Security, Nov. 2000.

[22] Archipelago: An Island-Based File System for Highly Available and Scalable Internet Services. USENIX Windows Systems Symposium, August 2000.

[23] Proof-Carrying Authentication. Andrew W. Appel and Edward W. Felten. Proc. of 6th ACM Conference on Computer and Communications Security, Nov. 1999.

[24] An Empirical Study of the SHRIMP System. Matthias A. Blumrich, Richard D. Alpert, Yuqun Chen, Douglas W. Clark, Stefanos, N. Damianakis, Cezary Dubnicki, Edward W. Felten, Liviu Iftode, Margaret Martonosi, Robert A. Shillner, and Kai Li. Proc. of 25th International Symposium on Computer Architecture, June 1998.

[25] Performance Measurements for Multithreaded Programs. Minwen Ji, Edward W. Felten, and Kai Li. Proc. of 1998 SIGMETRICS Conference, June 1998.

[26] Understanding Java Stack Inspection. Dan S. Wallach and Edward W. Felten. Proc. of 1998 IEEE Symposium on Security and Privacy, May 1998.

[27] Extensible Security Architectures for Java. Dan S. Wallach, Dirk Balfanz, Drew Dean, and Edward W. Felten. Proc. of 16th ACM Symposium on Operating Systems Principles, Oct. 1997. Outstanding Paper Award.

[28] Web Spoofing: An Internet Con Game. Edward W. Felten, Dirk Balfanz, Drew Dean, and Dan S. Wallach. Proc. of 20th National Information Systems Security Conference, Oct. 1997.

[29] Reducing Waiting Costs in User-Level Communication. Stefanos N. Damianakis, Yuqun Chen, and Edward W. Felten. Proc. of 11th Intl. Parallel Processing Symposium, April 1997.

[30] Stream Sockets on SHRIMP. Stefanos N. Damianakis, Cezary Dubnicki, and Edward W. Felten. Proc. of 1st Intl. Workshop on Communication and Architectural Support for Network-Based Parallel Computing, February 1997. (Proceedings available as Lecture Notes in Computer Science #1199.)

[31] Early Experience with Message-Passing on the SHRIMP Multicomputer. Richard D. Alpert, Angelos Bilas, Matthias A. Blumrich, Douglas W. Clark, Stefanos Damianakis, Cezary Dubnicki, Edward W. Felten, Liviu Iftode, and Kai Li. Proc. of 23rd Intl. Symposium on Computer Architecture, 1996.

[32] A Trace-Driven Comparison of Algorithms for Parallel Prefetching and Caching. Tracy Kimbrel, Andrew Tomkins, R. Hugo Patterson, Brian N. Bershad, Pei Cao, Edward W. Felten, Garth A. Gibson, Anna R. Karlin, and Kai Li. Proc. of 1996 Symposium on Operating Systems Design and Implementation.

[33] Java Security: From HotJava to Netscape and Beyond. Drew Dean, Edward W. Felten, and Dan S. Wallach. Proc. of 1996 IEEE Symposium on Security and Privacy.

[34] Integrated Parallel Prefetching and Caching. Tracy Kimbrel, Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. Proc. of 1996 SIGMETRICS Conference.

[35] Software Support for Virtual Memory-Mapped Communication. Cezary Dubnicki, Liviu Iftode, Edward W. Felten, and Kai Li. Proc. of Intl. Parallel Processing Symposium, April 1996.

[36] Protected, User-Level DMA for the SHRIMP Network Interface. Matthias A. Blumrich, Cezary Dubnicki, Edward W. Felten, and Kai Li. Proc. of 2nd Intl. Symposium on High-Performance Computer Architecture, Feb. 1996

[37] Improving Release-Consistent Shared Virtual Memory using Automatic Update . Liviu Iftode, Cezary Dubnicki, Edward W. Felten, and Kai Li. Proc. of 2nd Intl. Symposium on High-Performance Computer Architecture, Feb. 1996

[38] Synchronization for a Multi-Port Frame Buffer on a Mesh-Connected Multicomputer. Bin Wei, Gordon Stoll, Douglas W. Clark, Edward W. Felten, and Kai Li. Parallel Rendering Symposium, Oct. 1995.

[39] A Study of Integrated Prefetching and Caching Strategies. Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. Proc. of 1995 ACM SIGMETRICS Conference. Best Paper award.

[40] Evaluating Multi-Port Frame Buffer Designs for a Mesh-Connected Multicomputer. Gordon Stoll, Bin Wei, Douglas W. Clark, Edward W. Felten, Kai Li, and Patrick Hanrahan. Proc. of 22nd Intl. Symposium on Computer Architecture.

[41] Implementation and Performance of Application-Controlled File Caching. Pei Cao, Edward W. Felten, and Kai Li. Proc. of 1st Symposium on Operating Systems Design and Implementation, pages 165-178, November 1994.

[42] Application-Controlled File Caching Policies. Pei Cao, Edward W. Felten, and Kai Li. Proc. of USENIX Summer 1994 Technical Conference, pages 171-182, 1994.

[43] Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer. Matthias A. Blumrich, Kai Li, Richard D. Alpert, Cezary Dubnicki, Edward W. Felten, and Jonathan S. Sandberg. Proc. of Intl. Symposium on Computer Architecture, 1994.

[44] Performance Issues in Non-Blocking Synchronization on Shared-Memory Multiprocessors. Juan Alemany and Edward W. Felten. Proceedings of Symposium on Principles of Distributed Computing, 1992.

[45] Improving the Performance of Message-Passing Applications by Multithreading. Edward W. Felten and Dylan McNamee. Proceedings of Scalable High-Performance Computing Conference (SHPCC), 1992.

[46] A Highly Parallel Chess Program. Edward W. Felten and Steve W. Otto. 1988 Conference on Fifth Generation Computer Systems.

Other Publications

[47] Freedom to Tinker weblog, at http://www.freedom-to-tinker.com. Commentary on technology law and policy. Approximately 4000 readers per day.

[48] Secure, Private Proofs of Location. Brent Waters and Edward W. Felten. Submitted for publication, January 2003.

[49] An Efficient Heuristic for Defense Against Distributed Denial of Service Attacks using Route-Based Distributed Packet Filtering. Michael A. Schneider and Edward W. Felten. Submitted for publication, January 2003.

[50] Written testimony to House Commerce Committee, Subcommittee on Courts, the Internet, and Intellectual Property, oversight hearing on "Piracy of Intellectual Property on Peer to Peer Networks." September 2002.

[51] Written testimony to Senate Judiciary Committee hearings on "Competition, Innovation, and Public Policy in the Digital Age: Is the Marketplace Working to Protect Digital Creativity?" March 2002.

[52] Informed Consent Online: A Conceptual Model and Design Principles. Batya Friedman, Edward W. Felten, and Lynette I. Millett. Technical Report 2000-12-2, Dept. of Computer Science and Engineering, University of Washington, Dec. 2000.

[53] Mechanisms for Secure Modular Programming in Java. Lujo Bauer, Andrew W. Appel, and Edward W. Felten. Technical Report CS-TR-603-99, Department of Computer Science, Princeton University, July 1999.

[54] A Java Filter. Dirk Balfanz and Edward W. Felten. Technical Report 567-97, Dept. of Computer Science, Princeton University, October 1997.

[55] Inside RISKS: Webware Security. Edward W. Felten. Communications of the ACM, 40(4):130, 1997.

[56] Simplifying Distributed File Systems Using a Shared Logical Disk.Robert A. Shillner and Edward W. Felten. Princeton University technical report TR-524-96.

[57] Contention and Queueing in an Experimental Multicomputer: Analytical and Simulation-based Results. Wenjia Fang, Edward W. Felten, and Margaret Martonosi. Princeton University technical report TR-508-96.

[58] Design and Implementation of NX Message Passing Using SHRIMP Virtual Memory Mapped Communication. Richard D. Alpert, Cezary Dubnicki, Edward W. Felten, and Kai Li. Princeton University technical report TR-507-96.

[59] Protocol Compilation: High-Performance Communication for Parallel Programs. Edward W. Felten. Ph.D. dissertation, Dept. of Computer Science and Engineering, University of Washington, August 1993.

[60] Building Counting Networks from Larger Balancers. Edward W. Felten, Anthony LaMarca, and Richard Ladner. Univ. of Washington technical report UW-CSE-93-04-09.

[61] The Case for Application-Specific Communication Protocols. Edward W. Felten. Univ. of Washington technical report TR-92-03-11.

[62] A Centralized Token-Based Algorithm for Distributed Mutual Exclusion. Edward W. Felten and Michael Rabinovich. Univ. of Washington technical report TR-92-02-02.

[63] Issues in the Implementation of a Remote Memory Paging System. Edward W. Felten and John Zahorjan. Univ. of Washington technical report TR-91-03-09.