Abstract

The AFRL/Cornell Information Assurance Institute supports a broad spectrum of research aimed at developing a science and technology base to enhance information assurance and networked information systems trustworthiness—system and network security, reliability, and assurance. The institute also fosters closer collaborations between Cornell and AFRL researchers, as well as facilitating technology transfer and exposing Cornell researchers to problems facing the Air Force.
**Title and Subtitle**

AFRL/Cornell Information Assurance Institute

**Authors**

Schneider, Fred B.

**Performing Organization**

Cornell University  
4130 Upson Hall  
Ithaca, NY 14851

**Sponsoring / Monitoring Agency**

AFOSR/NM  
4015 Wilson Boulevard, Room 713  
Arlington, VA 22203-1954

**Abstract**

The AFRL/Cornell Information Assurance Institute supports a broad spectrum of research aimed at developing a science and technology base to enhance information assurance and networked information systems trustworthiness—system and network security, reliability, and assurance. The institute also fosters closer collaborations between Cornell and AFRL researchers, as well as facilitating technology transfer and exposing Cornell researchers to problems facing the Air Force.
1 Introduction

The AFRL/Cornell Information Assurance Institute (IAI) was established at Cornell by an initial grant from AFOSR in March 2000. IAI was created as a prototype for a new mode of funding. And, after 2 years, the results of this experiment confirm that this new funding mode—granting research funds to a University center having close geographic and intellectual proximity to, but loose affiliation with, an AFRL laboratory—has enormous leverage:

- IAI funding has enabled some absolutely first-rate Computer Science research to be performed at Cornell.
- IAI funding has facilitated interactions between Cornell researchers and AFRL staff, with research at Cornell now having clear relevance to the research needs of the Air Force.

Specific research accomplishments supported under the auspices of IAI are summarized below (§2); details can be found in the publications listed at the end of this report (§4). Technology transitions and DoD interactions are also discussed (§3). Figure 1 lists those researchers at Cornell (along with their specializations) whose work has been supported, in part, by IAI.

2 Summary of Research Accomplishments

Scalable Fault-tolerant Systems (Birman, van Renesse). This effort has focused on scalability of the publish-subscribe paradigm and has interacted extensively with Rome/AFRL researchers to understand specific issues arising from application of the publish-subscribe paradigm within the JBI effort and within other related military systems. The most significant accomplishments include the development of (i) the Astrolabe scalable monitoring and management framework and (ii) the Astrocast publish-subscribe structure based on a novel Bimodal Multicast. Together, these bring a new and remarkably flexible way of implementing publish-subscribe services with good scaling properties, stability under stress, and a high quality of security.

Program Refinement Logic (Constable, Kreitz). Program Refinement Logic (PRL) is a logical programming environment that provides substantial automation in the design, coding, verification, and evolution of large software systems. It is based on the latest version of Nuprl proof development system. A first prototype of a formal digital library of algorithmic knowledge (FDL) has been completed. FDL provides an infrastructure for
Kenneth Birman: Distributed computing, fault-tolerant network systems, distributed systems security, large-scale network applications.


Alan Demers: Database systems, database replication, and algorithms.

Johannes Gehrke: Database systems and data mining.

Joseph Y. Halpern: Reasoning about knowledge and uncertainty, distributed computing, security.

Dexter Kozen: Proof carrying code, program logics, and semantics.

Christoph Kreitz: Applied logic, automated reasoning, software assurance.

J. Gregory Morrisett: Programming languages, compilers, distributed systems, language-based security.

Andrew Myers: Programming languages, security, mobile code.

Robbert Van Renesse: Distributed computing, fault-tolerant network systems, distributed systems security, large-scale network applications.

Fred. B. Schneider: Distributed systems security and fault-tolerance, mobile code, concurrent programming.

Emin Gun Sirer: Secure distributed systems, extensible operating systems, language-based security, automated testing.

Jayavel Shanmugasundaram: Internet data management, database systems, and query-processing in emerging system architectures.

Figure 1: IAI Staff and Research Interests
verifying and synthesizing software systems by supporting the creation of certified algorithmic knowledge, the cooperation of multiple theorem proving systems, and flexible yet controlled access to the archived knowledge. Users may contribute library contents using the Nuprl, MetaPRL, JProver, and PVS theorem provers.

FDL has been used to support:

- code transformations that improve performance and enable protocols to be made adaptive while preserving functionality in connection with a self-adaptive task allocation manager to control processing of real-time media over a network through coordinated local schedules,
- the creation of formal courseware, and
- the translation of formal proofs into natural language.

**Databases and Data Mining (Gehrke).** The Cougar Project has produced database technology to support distributed wireless sensor networks with millions of nodes. Here, novel distributed query processing strategies for long-running queries permit in-network aggregation and can trade communication for local computation, increasing the lifetime of the network by up to an order of magnitude. A first version of the system was demonstrated at 29 Palms in California (Fall 2001) and the ACM Sigmod Conference (2002).

The Himalaya Project has created some of the world’s fastest data mining algorithms for mining long itemsets, classification tree construction, and also regression-tree construction and sequence mining. Other work focused on pushing user-defined constraints (such as defined by intrusion-detection systems) deep into the mining algorithm, in order to improve performance by orders of magnitude versus simple a-priori model construction and a-posteriori model pruning via constraints. This project also investigated privacy-preserving data mining algorithms, where datasets can be shared publicly without compromising values of individual records while at the same time ensuring that accurate statistical summary information can still be recovered.

**Formalizing Security (Halpern).** The use of modal logic has led to a new formalization of secrecy. This formalization is consistent with Sutherland’s notion of nondeducibility, subsumes both separability generalized non-interference and nondeducibility on strategies, and is able to handle
probabilistic secrecy, resource-bounded reasoning, as well as downgrading of information.

In addition, a new first-order logic was developed for reasoning about security policies. The formalism is a fragment of first-order logic that can both express many policies of interest and is tractable. Based on the logic, a prototype reasoning engine has been designed. Its user interface is intended for non-logicians, allowing them to enter policies, facts about principals, and then to ask questions about the policies.

Avoiding Malicious Boot Firmware (Kozen). In collaboration with Architecture Technologies Corporation (Ithaca, NY) and CodeGen Inc. (Palo Alto, CA) a prototype certifying compiler and verifier for detecting malicious boot firmware has been developed. Boot firmware modules are automatically verified against a standard security policy, as they are loaded. Among other things, the security policy being enforced asserts that drivers must access other devices only through a strict interface and may not access memory or bus addresses not allocated to them. Efficient Code Certification, along with inexpensive static checks on the compiled code, suffice to guarantee dynamic properties of the program at run time. The prototype is compliant with the now widely used IEEE 1275 Open Firmware standard for boot firmware. Sample device drivers written in Java for a block-oriented storage device and a PCI bus have been successfully compiled.

Cyclone Compiler (Morrisett). Cyclone is type-safe programming language that can be roughly characterized as a “superset of a subset of C.” The type system of Cyclone accepts many C functions without change and uses the same data representations and calling conventions as C for a given type constructor. It also rejects many C programs to ensure safety. For instance, it rejects programs that perform (potentially) unsafe casts, that use unions of incompatible types, that (might) fail to initialize a location before using it, that use certain forms of pointer arithmetic, or that attempt to do certain forms of memory management. All of the analyses used by Cyclone are local (i.e., intra-procedural) to enable scalability and separate compilation. The analyses are carefully constructed to avoid unsoundness in the presence of threads.

Experimental validations of Cyclone show great promise. For systems applications, such as a simple web server, Cyclone introduces virtually no overhead at all. This is not surprising, as these applications tend to be I/O-bound. For scientific applications, a much larger overhead is seen (around
5x for a naive port, and 3x with an experienced programmer). Some of that overhead is due to bounds and null pointer checks on array access, which can be eliminated using intra-procedural analysis; other overhead arises from the use of “fat pointers” and the fact that GCC does not always optimize struct manipulation.

**Secure Program Partitioning (Myers).** The Jif/split prototypes bring a new means to ensure that data confidentiality and integrity are preserved in distributed systems in spite of untrusted hosts and mutually distrusting principals. This problem is particularly relevant to information systems used by mutually distrusting organizations, such as the dynamic coalitions that arise in military settings. With Jif/split, programs are automatically partitioned into communicating subprograms that run on the available, partially trusted hosts; to protect data integrity, information and code are also replicated across the available hosts. If any host is subverted, then only principals that have explicitly stated trust in that host need fear a violation of confidentiality.

**Inlined Reference Monitors (Schneider, Morrisett).** In-lined reference monitors are a new approach to implementing traditional reference monitors whereby a desired end-to-end security policy is formulated using a high-level declarative policy language and then a rewriting tool is used to automatically rewrite untrusted code into code that respects the policy. The rewriting tool works by inserting extra state and dynamic checks into the untrusted code so that the code becomes self-monitoring.

Having developed prototypes for Intel X86 and Java JVM, the central question is one of practicality. To this end:

- A set of kernel modifications was developed to support a prototype IRM rewriter in Microsoft’s Windows operating system.
- A prototype MSIL (Microsoft Intermediate language) IRM realization has been developed. It implements an aspect-oriented programming metaphor for MSIL assembly language (rather than for a high-level language).

**Internet Data Management and Retrieval (Shanmugasundaram).** The QUARK project aims to integrate the database and information retrieval worlds by building a next-generation database system for handling both structured and unstructured data. This has required the development
of new techniques for storing and querying semi-structured data (containing a mix of structured and unstructured data) by using structured relational database systems. Techniques have also been developed for evaluating exploratory ranked keyword search queries over semi-structured data.

The PEPPER project has two main goals:

- to build an efficient information dissemination (or publish-subscribe) system for large-scale distributed systems and
- to develop a query processing layer for peer-to-peer networks.

The first goal required the development of a new, scalable index structure, called RPH-trees, for indexing user preferences so that only the relevant users are notified when new information becomes available. An interesting feature of RPH-trees is that they dynamically adapt to the information workload. For example, if there is a sudden burst of information about vehicle movement in Northern Afghanistan, the RPH-tree dynamically and automatically adjusts itself so that this information is processed efficiently and without delay. For the second goal, a fault-tolerant and scalable peer-to-peer index structure called P-trees has been developed. P-trees can support range queries in addition to equality queries.

**MagnetOS (Sirer).** MagnetOS is a new distributed operating system for ad hoc networks. It extends the effective lifetime of an ad hoc or sensor network through dynamic object migration, providing a single system image of a unified Java virtual machine across the nodes comprising an ad hoc network. By automatically and transparently partitioning applications into components and dynamically placing these components on nodes within the ad hoc network, MagnetOS reduces energy consumption, avoids hotspots, and increases system longevity—system longevity is increased by a factor of four to five, in fact.

Developing MagnetOS required solving two significant problems in ad hoc networks:

- **Multipath route selection.** Most routing algorithms—including those that are used in the core of the Internet—use single-path routing and thus are slow to respond to failures and frequently suffer path failures. Consequently, a new, efficient algorithm for constructing highly-reliable path sets has been developed.

- **Hybrid Routing Framework.** Traditional routing algorithms either proactively disseminate route updates or defer route discovery
until needed by a client. Choosing between the two regimes is difficult, since the tradeoff changes based on node mobility rate and communication patterns. This has led to the development of a new family of routing protocols that combine proactive and a reactive routing algorithms. These new protocols automatically adjust the radius of proactive information dissemination to discover routes with low overhead and latency.

3 DoD Interactions and Technology Transitions

A variety of technology transitions and interactions with DoD organization occurred during the period of this funding:

- Schneider chaired a study for DARPA IPTO Program Manager Jay Lala on promising research directions for Self-Healing Networked Information Systems.

- Schneider chaired the DARPA IPTO Oasis Dem-Val External Evaluation Committee.

- Morrisett and Schneider worked with Microsoft to develop a .NET in-lined reference monitor (IRM).

- Researchers at Carnegie-Mellon University, Princeton University, University of California (Riverside), University of Newcastle-Upon-Tyne, and Intel Research are all now building on PoET/PSLang IRM tools developed by Schneider and collaborators.

- Further public releases of the Jif compiler have been made available at the Jif web site, http://www.cs.cornell.edu/jif. The Jif language extends the Java programming language with support for information flow control. The Jif compiler is implemented on top of the Polyglot extensible compiler framework for Java. The Polyglot framework has also been released publicly at http://www.cs.cornell.edu/projects/polyglot, and researchers at Princeton University are using this framework in their own research. The releases of both Jif and Polyglot are provided as Java source code and work on Unix and Windows platforms.

- AT&T research is collaborating to develop the Cyclone language, compiler, and tools. In addition, researchers at the University of Maryland, the University of Utah, Princeton, and the University of Pennsylvania, and Cornell are all using Cyclone to develop research prototypes.
4 Publications Supported under this Grant


(37) A. Dobra and J. E. Gehrke. SECRET: A Scalable Linear Regression Tree Algorithm. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Edmonton, Alberta, Canada, July 2002).


(41) A. Faradjian, J. E. Gehrke and P. Bonnet. GADT: A Probability Space ADT For Representing and Querying the Physical World. Proceedings of the 18th International Conference on Data Engineering (ICDE 2002) (San Jose, California, February 2002).


(45) V. Ganti, J. E. Gehrke, and Raghu Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. IEEE Transactions on Knowledge and Data Engineering 13, 1 (January/February 2001), 50–63.


(83) D. Kozen. Computational inductive definability. Submitted for publication.


(86) D. Kozen. Some results in dynamic model theory (abstract). In E. A. Boiten and B. Möller, editors, Proceedings Conference on Mathematics of Program Construction (MPC’02), Lecture Notes in Computer Science, Volume 2386 (July 2002), 21.


(129) A. Sabelfeld and A. C. Myers. End-to-end security via program analysis. Submitted for publication.


(137) R. van Renesse, and K. Birman. Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining. Submitted to ACM TOCS.


(146) Z. Xiao, R. van Renesse and K. Birman. Optimizing Buffer Management for Reliable Multicast. *Dependable Systems and Networks (DSN ’02)* (Bethesda, Maryland, July 2002), IEEE.


