# Summary: the NSF Workshop on Formal Methods for Security

Stephen Chong
*Harvard University*

Joshua Guttman
*Worcester Polytechnic Institute and MITRE*

Anupam Datta
*Carnegie Mellon University*

Andrew Myers
*Cornell University*

Benjamin Pierce
*University of Pennsylvania*

Patrick Schaumont
*Virginia Tech*

Tim Sherwood
*University of California, Santa Barbara*

Nickolai Zeldovich
*Massachusetts Institute of Technology*

August 2, 2016

**Abstract**

We briefly summarize the report of the NSF Workshop on Security and Formal Methods, 19–20 November 2015. For the full report, see `https://arxiv.org/abs/1608.00678`.

Cybersecurity is everyone's problem. The target may be the electric grid, government systems storing sensitive personnel data, intellectual property in the defense industrial base, or banks and the financial system. Adversaries range from small-time criminals to nation states and other determined opponents who will explore an ingenious range of attack strategies. And the damage may be tallied in dollars, in strategic advantage, or in human lives. Systematic, secure system design is urgently needed, and we believe that rigorous *formal methods* are essential for substantial improvements.

Formal methods enable reasoning from logical or mathematical specifications of the behaviors of computing devices or processes; they offer rigorous proofs that all system behaviors meet some desirable property. They are crucial for security goals, because they can show that no attack strategy in a class of strategies will cause a system to misbehave. Without requiring piecemeal enumeration, they rule out a range of attacks. They offer other benefits too: Formal specifications tell an implementer unambiguously what to produce, and they tell the subsequent user or integrator of a component what to rely on it to do. Since many vulnerabilities arise from misunderstandings and mismatches as components are integrated, the payoff from rigorous interface specifications is large.

Adoption of formal methods in various areas (including verification of hardware and embedded systems, and analysis and testing of software) has dramatically improved the quality of computer systems. We anticipate that formal methods can provide similar improvement in the security of computer systems.

Moreover, formal methods are in a period of rapid development and significantly broadening practical applications. While formal methods have long been associated with cybersecurity

applications, new techniques offer deeper evidence for security goals across a wider range of components, and for the systems built from them.

Without broad use of formal methods, security will always remain fragile. Attackers have a clear advantage in what is currently a match between the cleverness of the attacker and the vigilance of the defender. Formal methods provide guidance for gapless construction, and for checking that an artifact has no points of entry for the adversary. Formal methods always use models, and thus can exclude only gaps that are expressible in those models. However, each model has specific, well-defined assumptions, which help focus a security analyst's attention on whether the actual system satisfies these properties.

The NSF workshop on Security and Formal Methods, held 19–20 November 2015, brought together developers of formal methods, researchers exploring how to apply formal methods to various kinds of systems, and people familiar with the security problem space. Participants were drawn from universities, industry research organizations, government, and a selected pool of scientists from foreign institutions. We explored how current research results and strategies can provide improved secure systems using contemporary formal methods, and how these goals can shape future refinements to formal methods.

The workshop was organized into four main areas: (i) *Hardware architecture*, (ii) *Operating systems*, (iii) *Distributed systems*, and (iv) *Privacy*. Each area had an expert area chair (or pair of chairs), who guided discussion and helped to write a section of the report below. Participants were assigned to an area for part of the workshop, with whole group sessions and cross-cutting groups to consider interactions among abstraction layers. These discussions led to the following observations, conclusions, and recommendations:

1. Formal methods for security will have an enormous effect in the coming years. Recent advances now enable their use at scales that were previously impossible. The resulting security improvements will spur new investments in formal tools and techniques. This interplay will produce a virtuous circle of capital investments in the methods and increases in both the quality of secure systems and the productivity of security-minded developers.

2. Formal methods are the *only* reliable way to achieve security and privacy in computer systems. Formal methods, by modeling computer systems and adversaries, can prove that a system is immune to entire classes of attacks (provided the assumptions of the models are satisfied). By ruling out entire classes of potential attacks, formal methods offer an alternative to the "cat and mouse" game between adversaries and defenders of computer systems.

   Formal methods can have this effect because they apply a scientific method. They provide scientific foundations in the form of precise adversary and system models, and derive cogent conclusions about the possible behaviors of the system as the adversary interacts with it. This is a central aspect of providing a science of security.

3. "Formal methods for security" should be construed broadly, beyond just mechanized logical specifications and proofs. Formal methods include approaches to reasoning about computational entities in which logical or mathematical descriptions of the entities entail reliable conclusions about their behavior. Contemporary cryptography relies on formal methods in this broad sense, as does synthesis of secure programs and other correct-by-construction mechanisms. The broad notion is also particularly relevant for privacy, where formal methods naturally extend to rigorous statistical and causal analysis methods.

4. Stark challenges remain. Computer systems are built in *layers* (e.g., hardware, operating systems, applications, networking, and distributed algorithms) where each layer is typically built under the assumption that lower layers behave correctly and securely. Security may fail at all layers. Frequently, failure is due to mismatches between adjacent layers, when behaviors of a lower layer do not satisfy the assumptions of a higher layer. Moreover, different systems (or different stakeholders in a system) may seek different *security goals*. While traditional goals such as authentication and confidentiality are already hard to pin down precisely, privacy goals govern the conflict between data subjects who do not want information about them disclosed, versus data owners seeking useful or lucrative uses for the data.

5. There is no single set of "right" security and privacy guarantees for computer systems. The desired security and privacy guarantees may ultimately depend on specifics of the computer application and system deployment. This heightens the need to explore security guarantees rigorously, and particularly privacy guarantees. Privacy should be studied as part of a larger research program on personal data protection that encompasses fairness, transparency, and accountability. Hence, formal methods researchers should work with researchers in philosophy, law, public policy, and related disciplines to forge comprehensive privacy foundations and meaningful tools for protecting privacy.

6. The are many open and compelling research problems, including: (1) *Whole-system guarantees*: How to specify and ensure the security of a whole system (as opposed to individual components or abstraction layers within a system)? How can this be done while still enabling modular development and compositional reasoning? (2) *Abstractions:* What are the right abstractions to enable formal methods for security, including abstractions to present to the programmer and abstractions provided by the operating system and architecture? (3) *(In)Compatibility of Tools, Proofs, and Specifications*: To what extent can existing and new tools and techniques be standardized to enable compatibility of specifications, proofs, and interoperability of tools? (4) *Software Development and Formal Methods for Security*: How can formal methods for security be supported throughout the lifecycle of software and hardware? (5) *Transition to Practice*: What is required to enable formal methods for security at industrial scales and make them compatible with common industry processes?

7. Challenge problems have the potential to ignite the imagination and enthusiasm of the community and to stimulate research that pushes the boundary of what is possible using formal methods to secure computer systems. We propose several challenge problems, including the following:

    - *Develop a formally verified crypto-currency wallet.*
    - *Develop an end-to-end secure messaging system on a peer-to-peer overlay.*
    - *Develop privacy-preserving tools for scientific discovery (data exploration and analysis) by medical researchers, social scientists, and other academics working in data-intensive fields for daily work.*
    - *Verify functional correctness of a POSIX-like operating system.*
    - *Use the results to design a post-POSIX operating system offering assured security services.*

8. Security and formal methods are both relevant to a broad cross-section of the Computer Science curriculum. In undergraduate education, security problems should be discussed in a variety of courses in which they naturally arise. Rigorous techniques should be introduced relatively early in the curriculum, and connected with numerous activities which repay their use. Graduate education can follow suit at a more sophisticated level.

9. Usable tools and infrastructure are critical to formal methods for security. The community should encourage their development, refinement, and shared use. Possible ways to do so include the active encouragement by conferences and journals of the submission or evaluation of artifacts for formal methods for security, and the establishment of repositories of formal artifacts and security-relevant benchmarks and test suites.

10. Clean slate redesigns can liberate innovative, high-quality work, but most systems will use much existing infrastructure. A balance of both types of work is needed, to provide formal methods a clean shot at improving security, as well as a path to broad impact by local improvements in existing components.

Thus, we recommend both foundational scientific work and more applied engineering as foci for improving cybersecurity via formal methods.

## Acknowledgments

We thank the National Science Foundation for sponsoring the workshop and Program Managers Nina Amla and Anindya Banerjee for advice and discussions. We thank Michael Hicks for hosting the workshop at UMD College Park, and Tina Knight for administrative and logistical support. We thank Owen Arden, Tej Chajed, Justin Hsu, and Joseph McMahan for scribing the workshop discussions. We sincerely thank the workshop participants. Their active and enthusiastic engagement produced the core results of this report. The participants were: Nina Amla, Owen Arden, Anindya Banerjee, Tej Chajed, Steve Chong, John Criswell, Anupam Datta, Steven Drager, Nate Foster, Matt Fredrikson, Sol Greenspan, Carl Gunter, Aarti Gupta, Joshua Guttman, Michael Hicks, Justin Hsu, Bart Jacobs, Somesh Jha, Frans Kaashoek, Daniel Kifer, Boris Köpf, Rao Kosaraju, Shriram Krishnamurthi, Petros Maniatis, Z. Morley Mao, David Mazières, Jon McCune, Patrick McDaniel, Bill McKeever, Joseph McMahan, Ken McMillan, Toby Murray, Andrew Myers, Bryan Parno, Benjamin Pierce, Phil Regalia, Patrick Schaumont, Deborah Shands, Zhong Shao, Tim Sherwood, Elaine Shi, Cynthia Sturton, Jakub Szefer, Cesare Tinelli, David Walker, Xi Wang, and Nickolai Zeldovich.