

RESEARCH STATEMENT

ARNAB ROY

I am broadly interested in computer and network security, cryptography, programming language theory and logic. My current research has been on the confluence of these areas encompassing both theory and practice. I have used programming language techniques to model distributed network security protocols, logic to express and prove security properties of the protocols and cryptographic techniques to establish the soundness of the logical axiomatization. I have worked on practical application of a logic to express authorization policies and made theoretical advances to the framework as a result of the experience gained. While the topics of my current research have significant potential of further exploration, I am also excited about researching new areas in computer science in the future.

SUMMARY OF ACCOMPLISHMENTS

Security Analysis of Network Protocols. Present-day internet users and networked enterprises rely on key management and related protocols that use cryptographic primitives. In spite of the staggering financial value of, say, the total number of credit card numbers transmitted by SSL/TLS in a day, we do not have correctness proofs that respect cryptographic notions of security for many of these relatively simple distributed programs. The lack of assurance despite critical usage has led to many efforts to develop and use methods for proving security properties of network protocols. This area of research has had two important but historically independent foundations, one based on logic and symbolic computation, and one based on computational complexity theory. The symbolic approach, which uses a highly idealized representation of cryptographic primitives, has been a successful basis for formal logics and automated tools. Conversely, the computational approach yields more insight into the strength and vulnerabilities of protocols, but is more difficult to apply.

We have developed Protocol Composition Logic (PCL), a logic for proving security properties of network protocols in the symbolic model of cryptography. The logic is designed around a process calculus with actions for possible protocol steps including generating new random numbers, sending and receiving messages, and performing decryption and digital signature verification actions. A provable assertion holds in any protocol run despite arbitrary actions by a malicious adversary. PCL supports compositional reasoning about complex security protocols and we have applied it to a number of industry standards including SSL/TLS, IEEE 802.11i and Kerberos V5.

To provide stronger assurances - guaranteeing security in the computational model of cryptography - we developed Computational PCL, retaining similar syntax and proof system as PCL. Authentication properties of protocols are typically trace properties. For example, we can figure out whether two principals engaging in a protocol session are authenticated to each other or not by just looking at the trace of the protocol execution - actions performed by each principal and messages exchanged. Secrecy properties, on the other hand, are not trace properties in the computational model of cryptography. Computational indistinguishability, for example, requires that no computationally bound observer can feasibly distinguish a situation in which a secret is transmitted from a situation in which some non-informative values are transmitted instead. If we look at a single trace, this gives no real information about how likely an observer is to succeed. Instead, we must look at the probability distribution on traces, and determine the probability of success over the entire distribution. The nature of this property presents a challenge for proving computational

secrecy properties of protocols, since trace-based properties are naturally amenable to induction, while non-trace-based properties are not.

As part of work towards my PhD thesis, I made the following theoretical contributions to the security analysis of network protocols:

- (1) I formalized a form of inductive reasoning about secrecy in a set of new axioms and inference rules that are added to PCL and proved soundness of the system over a conventional symbolic protocol execution model.
- (2) I developed foundations for inductive analysis of computational security properties by proving connections between selected trace properties of protocol executions and non-trace complexity theoretic properties standard in the literature.
- (3) I formalized the aforesaid inductive properties in a set of new axioms and inference rules that are added to Computational PCL and proved soundness of the system over a standard cryptographic model with a probabilistic polynomial time adversary.

On the practical side, I have applied the above techniques to the analysis of widely used network security protocols:

- (1) Kerberos V5 is used for authentication and key exchange by many systems like Microsoft Windows, Stanford WebAuth and so on. I proved authentication and secrecy properties of the Kerberos V5 protocol with both symmetric-key and public-key initialization in both the symbolic and the complexity theoretic models. The proofs of key secrecy of Kerberos V5 in the complexity theoretic model are first in the literature.
- (2) I discovered a deficiency in Kerberos V5 with Diffie-Hellman initialization and suggested simple corrective measures. For the first time in the literature, I proved secrecy and authentication properties of this mode of Kerberos in the complexity theoretic model, thus constructing proofs of security of all three modes of Kerberos V5 in a uniform formal framework.
- (3) IKEv2 is the protocol used to set up a security association (SA) in the IPsec (used, for example, for VPN) protocol suite. I proved secrecy and authentication properties of the IKEv2 protocol in the complexity theoretic model for the first time in the literature.
- (4) IEEE 802.11i, marketed as WPA2 (WPA is Wi-Fi Protected Access), is a standard for secure wireless networking. Along with colleagues, I have proved authentication and key secrecy properties of IEEE 802.11i in the symbolic model for the first time in the literature.

The formal logic framework that I have developed have also been used by other researchers to prove the security properties of complex industrial protocols, such as researchers at Motorola Corporation who analyzed the security of the IEEE 802.11s standard, which is tasked to provide ways of establishing and securing a wireless mesh network.

Along with colleagues, I have also reviewed and analyzed the following industry track security protocols:

- IEEE 802.16e - Standard for local and metropolitan area networks: Air interface for fixed and mobile broadband wireless access systems.
- IEEE 802.11af - Authenticated key agreement for MACsec (Media Access Control security).
- IEEE 802.11r - Fast BSS (Basic Service Set) transition for IEEE 802.11.
- EMU EAP GPSK - Generalized Pre-shared Key Authentication method.
- Binding Update and Fast handover protocols in Mobile IPv6.

The analyses consisted of identifying flaws or deficiencies as well as providing assurance through formal proofs. In many cases, our recommendations were followed for improving the protocols. In addition, our analyses led to precise articulation and documentation of the security properties to be expected of the protocols.

Authorization Policies. Distributed Knowledge Authorization Logic (DKAL) is a new declarative authorization language for distributed systems developed by Yuri Gurevich and Itay Neeman; it is based on existential fixed-point logic and is considerably more expressive than existing authorization languages in the literature. We worked on the first practical application of DKAL, which was to the problem of automating source asset management (SAM) at Microsoft.

A large software company has many partners, contractors and subcontractors who need to access the sources of the various software products produced by the company. The ever-growing number of such requests necessitates clear access control policies regulating who is entitled to use what sources, where, for how long, and so on. The company also needs processes in place to efficiently implement those policies. While DKAL proved itself sufficiently expressive as well as convenient and elegant, the lessons we learned necessitated us to define and analyze operational aspects of the framework.

Two of the most important lessons we learned working on SAM were:

- (1) To separate the access control policy from the workflow. The workflow includes the procedural aspects of access decisions. The workflow can be quite complex to ensure the flexibility of the way access decisions are made, but the policy should be succinct. Furthermore, the policy should guarantee important safety properties of the access decision process for any workflow, eliminating the need to verify large amount of procedural code.
- (2) In the case of automated access control it is crucial for auditing purposes to properly log all human judgments. No set of policies and processes makes industrial access control deterministic. The inherent non-determinism is resolved by human judgments in making and delegating access decisions. Logging these judgments is necessary for auditing purposes as it allows the auditor to understand who made what decisions and whether they had the right to do so (either originally or by delegation).

As a result of the experience gained, we made some theoretical advances to DKAL. We separated policy and workflow and defined clear operational semantics. Logging the right information is a natural byproduct of the way operational semantics is defined. In addition we obtained some theoretical results characterizing the complexity of analyzing natural properties of policies. We explored two communication models for defining workflows and proved that deciding reachability questions is possible in polynomial time in certain cases whereas deciding invariants is coNP-complete.

FUTURE DIRECTIONS

As a researcher, I am motivated by the application of existing mathematical theory to problems of industrial importance and conversely, developing new theoretical foundations to model and analyze practical problems.

While several exciting directions of research are foreseeable in network protocol security research, I am working on two promising problems in this area with interesting preliminary results. The first problem is to automate verification and generation of security proofs in PCL. We have successfully automated the verification of a large class of protocol properties using Prolog as an implementation tool and published the results recently. The second problem is to provide quantitative guarantees of security for network protocols, for example, how many times a particular key should be used, a measure which is extremely important to protocol designers. This is a concretization of the asymptotic guarantees provided by Computational PCL.

The automation of Source Asset Management is a long term research and development project at Microsoft. I had the pleasure of being actively involved in the starting off of this project. In the short duration that I had as an intern at Microsoft Research, I made foundational contributions to the architecture of the system and charted a long term vision of the project along with my mentor

and SAM members. Theoretical results related to the system have been recently published as an MSR Technical Report.

I have diverse interests in Computer Science and Mathematics outside the area of formal methods for security. During my undergraduate studies at IIT Kharagpur, I worked on design automation, specification and verification of computer architectures using logical and algorithmic techniques, leading to two ACM Transaction articles.

I have immensely enjoyed the social aspects of doing research with colleagues at IIT Kharagpur and Stanford University, while also savouring the intellectual delights of working independently. I look forward to a research environment which has a vibrant community that embraces exploration while working at the forefront of technology.