A Discussion on “Security Protocols over open networks and distributed Systems: Formal methods for their Analysis, Design, and Verification”
S. Gritzalis, D. Spinellis, and P. Georgiadis

Presented by Jack G. Nestell

Topics for Discussion

I. Introduction
II. Discussion on the different logics and methods of reasonings of Formal Methods
III. Formal Specification Languages
IV. Robustness Principles
V. Formal Methods for Protocol Design
VI. Questions
II. Conclusion

Objective:

The biggest problem is typically not so much solving an issue as it is trying to find the best and most appropriate solution to the problem.

My Goal:

Not so much to understand the syntax and semantics as to understand and explain the logic.

"One point of tension in many formal methods is that their languages (and methods of reasoning/logic) may be more suitable to one type of specification than to others."

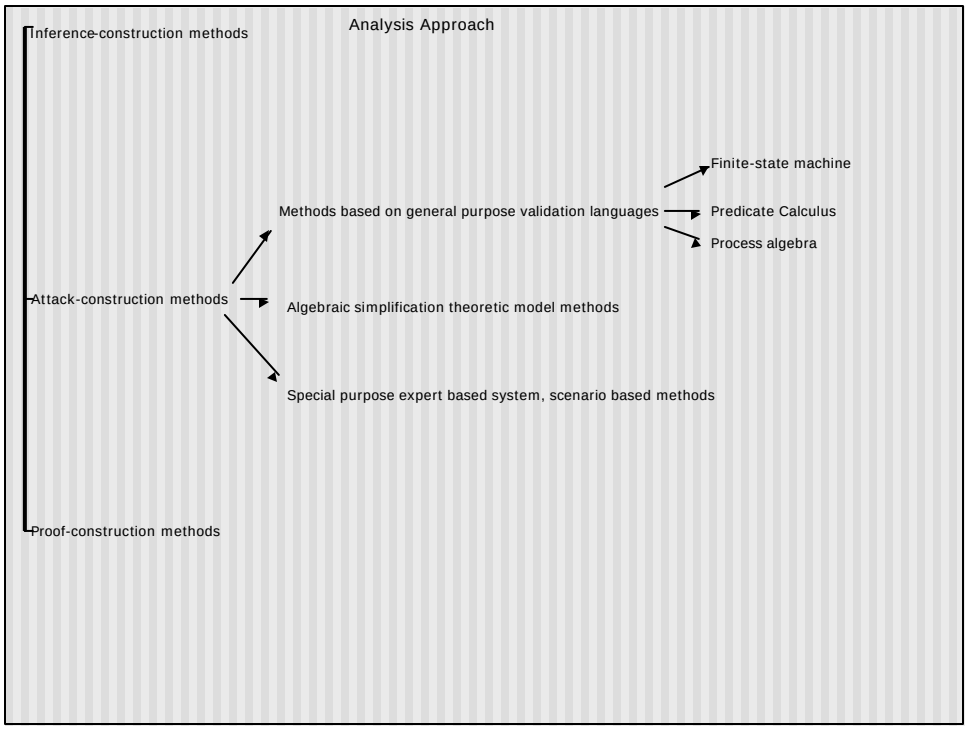
"An advocate of a particular formal method should tell potential users the method's domain of applicability… this starts with it's logic. Without knowing the proper domain of applicability, a user may inappropriately apply A formal method to an inappropriate domain"

Jeannette M. Wing "A Specifier's Introduction to Formal Methods"

The goal of my semester survey project is to elaborate upon three specific formal methods and to dicuss in depth The differeces in their logic and methods or reasoning

Logic and reasoning behind formal methods:

- Syntactic Domain, Semantic Domain, and Satisfies Relation
- Need to address pragmatic considerations. What domain is it for, what application, when is it used and how.
- All application domains are dynamic and unique



Attack-construction methods

- Construct probable attack sets based on the algebraic properties of the protocols algorithms.
- Targeted towards ensuring authentication, correctness, or security properties.
- They are not dependent on the correctness of a proposed logic.

- Main Disadvantage: The large number of possible events that must be examined.

Methods Based on general purpose validation languages:
"Generic" methods that analyze a security protocol as they would any other program whose correctness they Are trying to prove.

I. Finite-state machine:
II. Predicate Calculus:
There are many valid argument forms, however, that cannot be analyzed by truth functional methods, e.g., the classic syllogism: "All men are mortal. Socrates is a man. Therefore Socrates is mortal."
III. With-in a process algebra:
As the name suggests Process Algebra is a kind of algebra which is useful to describe and reason about process behaviours.
Underlying semantic foundation is transition systems. Systems defined in terms of states and transitions , where transitions are triggered by interactions.

Many notations and formalisms for Process Algebra have been defined. Ex: Calculus of Sequential Processes (CSP).

Attack-construction methods

Algebraic simplification theoretic model methods:

- Model a protocol with a collection of rules for transforming and reducing algebraic expressions representing messages.
- A formal method developed based on the algebraic term-rewriting properties of cryptographic systems.
- "We consider a protocol as a set of rules for passing messages between the participants" [4] p5
- Used to describe a model of a class of protocols and their possible attacks on those protocols as term-rewriting systems.
- Utilizes software tools based on a narrowing algorithm.
- NRL Protocol Analyzer
  There are a set of deterministic and nondeterministic rules which may apply under certain circumstances.
- Interrogator model
  uses a backward search strategy to construct path from insecure state to initial state.

Attack-construction methods...continued

Special Purpose expert system, scenario based methods:

- Protocol participants are modelled as communicating state machines whose messages to each other are intercepted by an intruder who can either destroy messages, modify them, or let them pass through unmodified.

Inference-construction methods

A logical inference system defines a consequence relation given in terms of a set of inference rules which map a set of well formed sentences in the specification language to a set of well-formed sentences.

- Taking a specification (what the desired results shall be) as a set of facts, one is able to derive new facts through the use of these inference rules.
- A formal method that allows one to predict a system's behavior without having to run or construct the protocol.
- Do not address secrecy
- Any fact is only a belief and does not need to be universal in time and space. BAN, presented by Burrows, Abadi, and Needham

I. Express the assumptions and goals as statements in a symbolic notation.
II. Transform the protocol steps into symbolic notation.
III. Apply a set of deduction rules.

BAN critiques:

I. Difficult to prove properties of BAN logic such as completeness. No complete semantics for the logic and the modeling of freshness.
II. Logic does not take into account the release of message contents and the interaction of the runs at different time of the same protocol.
III. The abstract level of BAN logic models results in difficult to assess hypotheses and protocol descriptions.

Proof-construction methods

- In order to overcome disadvantages of Inference and Attack construction methods, **Bolignano** propsed a method that targets *human-readable proofs*.
- Proofs- used as vulnerability or formal code inspections.
- Properties of problem are used to formalize the requirments and simplify the proofs.
- Places particular emphasis on the clear description of the problem
- Provides a clear separation between reliable and unreliable principals
- Automated within a framework of typed logics using the Coq proof assistant.
- Examples:
  Paulson
  Schneider
  Fabrega, Herzog, and Guttman (Strand Space Model)
  Snekkenes (HOL Theorem Prover)

Formal Specification Languages and tools for Automatically Analysing Cryptographic Protocols

- Discussed methods are usually not easily applied by analyst other than the developers
- Protocols have to be re-specified for each technique
- Design a single common protocol specification language that could be used as the input format for any formal analysis technique.

I. Meadows *heuristic* approach using NRL Protocol Analyzer
II. Brackin, Interface Specification Language (ISL) and Automatic Authentication Protocol Analyzer
III. Millen, Common Authentication Protocol Specification Language (CAPSL)

Robustness Principles

- Informal guidelines, useful independently of any logic.
- Complementary approach: utilize experience of good and bad practice into *empirical* rules. Relying or based on experiment or experience.
- Adherence to them adds to the simplicity and can help avoid confusions and mistakes.

✓ Be clear about security goals
✓ Be clear about purpose of encryption
✓ protocol does not make incorrect assumptions about cryptographic algorith
✓ Distinguish different protocl runs from each other
✓ Do not assume received message has only a particular form
✓ identity of a principal is essential to the meaning of a message, should be mentioned explicity in message
✓ sign before encrypting

- Boyd's intentional and extensional goals.
  Intentional goals- concerned with ensuring that the protocol runs correctly as specified.
  Extensional goals- concerned with what the protocol achieves for it's participants.
- Formal methods and structured design rules compliment each other.

Formal Methods for Protocol Design

- Design of secure protocols complex and difficult process
- Until recently secure protocol design was oriented towards analysis and verification of existing protocols.
- The next logical evolution of formal methods would be to design formal methods and tools to aid in the ex-ante protocol design. This implementation of formal methods into design can be accomplished several ways:

I. Use protocol design methods that cater well to formal methods analysis. Ex. Heintze and Tygar
II. Development of design principles that are used to develop protocols whose security is easier to evaluate. Ex: Gong and Syverson and notion of fail stop protocols.
III. Layered approach proposed by Meadows.
IV. Buttman, Staaman, and Wilhelm proposed the idea of channels.

V. Gollman, protocols error prone due to a language problem.

Questions

1) Explain the logic behind the NRL Protocol Analyzer that allows it to be used to prove security Properties of cryptographic protocols *as well as* locate security flaws.

The NRL Protocol Analyzer was designed to use narrowing to handle the fact that words obey reduction rules. In addition, it includes techniques and automatic support for using induction to prove that infinite sets of states are unreachable.

2) According to the authors (and based on the logic of BAN), one needs to "transform the protocol steps into symbolic notation". What is the reasoning behind transforming each protocol step into an idealized form according to BAN logic?

Informal notation is often ambiguous and obscure in its meaning. And therefore, it is not an appropriate basis for formal analysis. A message in an idealized form is essentailly a formula. In idealized form, parts messages that do not contribute to the beliefs of the recipient are omitted. Since BAN logic is an inference based method, this is critical. Idealized protocols are more complete specifications than the traditional descriptions.

Questions

3) In terms of the logic used by inference-construction methods, explain why inference-construction methods do not address secrecy.

The basic idea with inference-construction methods is that this method is based on the interaction between participants. Given certain inference rules, participants make decisions or form beliefs based on the participants they are communicateing with. Inference-construction methods address *authentication* (whether participants know who they are communicating with) and not *security* (whether information is revealed to those not meant to receive it.) This method is based on the idea that protocol ***participants*** can confidently reach desired conclusions. Authentication is based more on the the idea of interacting participants. Security is anayalzed by looking at the transmission and protection of data between participants which requires analysing the algebraic properties of the protocol itself. "Inference-construction methods concentrates on the beliefs of trustworthy parties involved in the protocols and on the evolution of these beliefs as a consequence of communication." [5]

Conclusion

- The three analysis methods discussed are useful at various levels of abstraction.
- More abstract models-used efficiently at earlier points in design stage.
- Combination of analysis methods may prove most comprehensive:

1)Initially-inference construction method, determine role of each message of protocol
2) Attack construction method-find simple attacks quickly.
3) Proof-construction method-investigate deeper properties.

- Or…..synthesis approach.
- Current research;

Bolignano-ITSEC evaluation
AAPA and CAPSL
Protocl suites to be used in commercial world

References

[1] “Security protoclols over open networks and distributed systems: Formal methods for their analysis, Design and verification” S. Gritzalis, D. Spinellis, and P. Georgiadis Computer Communications 22(8): 695-707, MAY 1999

[2] “A logic of Authentication”, Burrows, Abadi, and Needham SRC research report 39, Feb 28, 1989

[3] “A Logical Language for Specifying Cryptographic Protocol Requirements”, P. Syverson and C.Meadows Center for High Assurance Computer Systems, Naval Research Laboratory

[4]”Applying Formal Methods to the Analysis of a Key Management Protoclol”, Catherine Meadows, CSIT

[5]”Prudent Engineering Practice for Cryptographic Protocols”, M. Abadi and R. Needham, November 1, 1995

[6]”Modeling and verifying key-exchange protocols using CSP and FDR”A.W. Roscoe, Oxford University Computing Laboratory

[7]”Limitations on Design Principles for Public Key Protocols” P. Syverson, Center for High Assurance Computer Systems

[8]”Mechanized Proofs for a Recursive Authentication Protocol”, L.C. Paulson, University of Cambridge

[9]”A Specifier’s Introduction to Formal Methods”, J.M.Wing, Carnegie Mellon University

References

[10]”A HOL Extension of GNY for Automatically Analyzing Cryptographic Protocols” , Stephen H. Brackin, Arca Systems, Inc.

[11]”Programming Satan’s Computer”, R. Anderson and Roger Needham, Cambridge University Laboratory

[12]”Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR”, Gavin Lowe,Oxford University Computing Laboratory

[13]”The NRL Protocol Analyzer: An Overview”, C. Meadows, Center for High Assurance Computer Systems Naval Research Laboratory

[14]”Proving Properties of Security Protocols by Induction”,L.C. Paulson, University of Cambridge