Dagstuhl Seminar 01391: Specification and Analysis of Secure Cryptographic Protocols

David Basin
Universität Freiburg
Institut für Informatik
basin@informatik.uni-freiburg.de

Grit Denker Jon Millen
Computer Science Laboratory
SRI International
{denker,millen}@csl.sri.com

Gavin Lowe
Computing Laboratory
Oxford University
Gavin.Lowe@comlab.ox.ac.uk

Schloß Dagstuhl, September, 23–28, 2001

1 Motivation

Cryptographic protocols are the cornerstone of secure electronic communication, banking, and commerce. By providing functions like key management in distributed systems, individual and group authentication, anonymity, fair exchange, and policy negotiation, they support a spectrum of secure online activities such as financial account transactions, distributed sealed-bid auctions and their escrow services, voting, distributed and federated database access, and virtual private networks.

Designing cryptographic protocols is difficult. Cryptographic protocols are vulnerable to message modification attacks and it is surprisingly difficult to get even small protocols right. Moreover, their complexity is steadily increasing and it is nontrivial to compose or extend smaller protocols to more complex ones.

Formal methods have proven helpful for both cryptographic protocol design and analysis. The use of formal languages, including state machines, epistemic logics, and process algebras, supports the rigorous formalization of protocol models and their properties. Moreover, they provide a basis for using tools such as model checkers and theorem-provers to prove protocols correct or uncover security flaws.

The goal of this seminar is to bring together experts from the formal methods and security communities to study and compare existing modeling and analysis techniques and tools for cryptographic protocols, focusing on the following topics:

1. identifying and formalizing appropriate and practical security goals and security protocol analysis techniques,
2. classifying and comparing existing modeling techniques and formalisms, and
3. comparing existing tools, identifying the most urgent needs to integrate formal methods into real world protocol design, and suggesting future directions for tool design.

There are many questions and unresolved issues associated with each of these topics.

Regarding (1), security goals include message and entity authentication, secrecy, anonymity, integrity, non-repudiation, fair exchange, agreement, and denial of service resistance. There is currently no commonly accepted formal model of all of these and it is not even clear if this list is complete. Additional questions include: Which tool or analysis technique best supports which kind of goals? What is the relationship between formalizations of goals in different frameworks? Which protocol technique can handle "weak secrets," i.e., secrets that can be revealed by guessing? What security goals are essential in group communication applications? In this seminar we aim to identify commonly understood security goals and security protocol techniques and means to link goals with appropriate verification tools.

Regarding (2), currently many different techniques are used for modeling, e.g., event-based approaches using knowledge and belief logic abstractions, agent-based approaches modeling protocol processes using multiset rewriting, process algebra based approaches, and the strand space approach. How do these models differ? Natural candidates for a classification scheme are: synchronous versus asynchronous communication, complexity, decidability, practicability, which class of cryptographic protocols can be modeled (point-to-point, group communication, etc.), which cryptographic and other computations are supported, which analysis techniques are supported, and what is the scope, extensibility, and reusability of the modeling formalism. Heuristics and transformations play a role here too: how can protocols and their models be

safely transformed to bring them within the reach of current tools (e.g., reduced to small finite-state programs that can be model checked)?

Regarding (3), more sophisticated tools are required if protocol designers are to use them for real world problems. To what extent can tools scale that are based on (potentially infinite) state enumeration, finite state model checking, or automated (or interactive) theorem proving? In automated approaches, heuristics often play an important role in incorporating domain knowledge about particular protocols into the search process. Tool design issues include: how can the performance of heuristics be enhanced by exploiting knowledge about the protocol, message formats or attacker behavior in a rigorous way? Can heuristics be designed in a general, reusable style but still amenable to optimization? Finally, how can model checkers and theorem provers fruitfully interact with each other (e.g., the use of theorem provers to verify abstractions for model checking)?

2 List of participants

The following list of participants is alphabetically ordered.

Rafael Accorsi, University of Freiburg
(accorsi@informatik.uni-freiburg.de)

Roberto Amadio, University of Marseille
(amadio@cmi.uni-mrs.fr)

Alessandro Armando, University of Genova
(armando@dist.unige.it)

Tuomas Aura, Microsoft Research - Cambridge
(tuomaura@microsoft.com)

David Basin, University Freiburg
(basin@informatik.uni-freiburg.de)

Mike Bond, Cambridge University
(Mike.Bond@cl.cam.ac.uk)

Iliano Cervesato, ITT Inc. - Alexandria
(iliano@itd.nrl.navy.mil)

Ernie Cohen, Telcordia Technologies - Morristown
(ernie@research.telcordia.com)

Jorge R. Cuellar, Siemens - München
(Jorge.R.Cuellar@mchp.siemens.de)

Grit Denker, SRI - Menlo Park
(denker@csl.sri.com)

Nancy Durgin, Stanford University
(nad@cs.stanford.edu)

Michael Goldsmith, Formal Systems Ltd. - Oxford
(michael@fsel.com)

Dieter Gollmann, Microsoft Research - Cambridge
(diego@microsoft.com)

Dieter Hutter, DFKI Saarbrücken
(hutter@dfki.de)

Bart Jacobs, University of Nijmegen
(bart@cs.kun.nl)

Peeter Laud, University of Saarbrücken
(laud@cs.uni-sb.de)

Gavin Lowe, Oxford University
(Gavin.Lowe@comlab.ox.ac.uk)

Catherine Meadows, Naval Research - Washington
(meadows@itd.nrl.navy.mil)

Jon Millen, SRI - Menlo Park
(millen@csl.sri.com)

Sebastian Mödersheim, University of Freiburg
(moedersh@informatik.uni-freiburg.de)

David Monniaux, ENS - Paris
(David.Monniaux@ens.fr)

Lawrence Paulson, Cambridge University
(Larry.Paulson@cl.cam.ac.uk)

Andreas Pfitzmann, TU Dresden
(pfitza@inf.tu-dresden.de)

Michaël Rusinowitch, INRIA Lorraine - Nancy
(Michael.Rusinowitch@loria.fr)

André Scedrov, University of Pennsylvania
(scedrov@saul.cis.upenn.edu)

Vitaly Shmatikov, SRI - Menlo Park
(shmat@csl.sri.com)

Scott D. Stoller, SUNNY - Stony Brook
(stoller@cs.sunysb.edu)

Carolyn Talcott, Stanford University
(clt@cs.stanford.edu)

Luca Viganò, University of Freiburg
(luca@informatik.uni-freiburg.de)

Dennis Volpano, Naval Postgrad. School
(volpano@cs.nps.navy.mil)

Christoph Walther, TU Darmstadt
(chr.walther@informatik.tu-darmstadt.de)

Thomas Wilke, Universitt Kiel
(wilke@ti.informatik.uni-kiel.de)

Irfan Zakiuddin, Defence Evaluation Research Agency - Malvern
(i.zakiuddin@eris.dera.gov.uk)

3 Abstracts

Tuomas Aura

Title: (Why) are new security protocols still needed?

Abstract: The purpose of this talk was to ask why so many new security protocols are being designed, and whether that is really the right thing to do. Would it not be better for assurance to standardize one well-analyzed protocol, or a protocol family, which would then be used in all applications? The design methods and tools are already sufficient to guarantee that no major attack against the standard would be discovered.

The talk was mostly a survey of the reasons why new protocols are introduced. It turns out that many changes in the environment and requirements inevitably mean changes in the security protocols. New network and application architectures change the number of principals and the trust relations and connections between them. The protocols must perform optimally in a widening range of environments from the globally broadband Internet to low-end wireless links and mobile devices. No single security solution seems to cover all the requirements and security mechanisms have penetrated all layers of the protocol stack. Moreover, when the main goals of protocol design have been achieved, both attackers and protocol designers focus on other qualities like denial-of- service resistance, anonymity, usability and robustness of implementation.

It looks like instead of a single standard protocol, an evolving set of protocol design patterns or composable building blocks would be needed. They could potentially improve the quality of protocols, in particular when designed under strict time constraints by engineers who are not always security experts.

Title: A flaw in a denial-of-service resistant protocol

Abstract: The possibility of formally specifying and analyzing DoS attacks and resistance was contemplated after Cathy Meadows's talk on DoS. Since there is a shortage of concrete examples, I presented a real but difficult-to-detect flaw in an early version of a DoS- resistant protocol.

In the protocol, the client must solve a cryptographic puzzle before the server commits its memory and processing power to the authentication. In order to exhaust the server's resources, an attacker would need to solve large numbers of such puzzles, which would be too expensive. The problem is that while the protocol protects the server against

flooding attacks, the attacker can deny service for individual clients by solving the puzzles for them and submitting the solutions to the server before the client does. The server will remember the solved puzzles and ignore duplicate solutions sent later by the real client. The problem is solved by a minor modification of the puzzle. The flaw would probably have been spotted earlier in the design process, had there been a set of well-defined DoS-resistance requirements.

David Basin

Title: Lazy analysis of security protocols

Abstract: We present an approach to modeling security protocols using lazy data types in a higher-order functional programming language. Our approach supports the formalization of protocol models in a natural and high-level way, and the automated analysis of safety properties using infinite-state model checking, where the model is explicitly constructed in a demand-driven manner. We also discuss recent extensions of this work and the construction of a general purpose protocol analysis tool.

Mike Bond

Title: First steps in cryptoprocessor API analysis

Abstract: Cryptoprocessors are tamper-resistant processors designed to store and manipulate sensitive information. Users communicate with the cryptoprocessor through its "Security API". This talk introduces cryptoprocessor APIs as a target for formal analysis and verification. The talk first describes the typical properties of a cryptoprocessor, and illustrates these with the example of the Visa Security Module. A prototype analysis method is described – "Type System Analysis" – which successfully found an attack on the Visa Security Module. The talk discusses how the method might be extended to deal with larger problems. The IBM 4758 "Common Cryptographic Architecture" is introduced as a challenge, and the talk explores tools and methods that might be brought to bear to deal with a problem of this magnitude.

Iliano Cervesato

Title: The wolf within

Abstract: A formal specification of a security protocol cannot be limited to listing the messages exchanged. In MSR, each construct is associated with typing and data access specification (DAS) rules, which describe under which circumstances a principal can access keys and other information. A protocol specification is completed with a description of the intruder in the style of Dolev and Yao. In this extended abstract, we

show that the protocol determines the intruder: more precisely we show that the Dolev-Yao intruder rules can be automatically reconstructed from the DAS rules, and that the DAS rules can themselves be inferred from annotated typing declarations for the various message constructors.

Ernie Cohen

Title: A simple trick to avoid junk

Abstract: We show a simple transformation for cryptographic protocols that preserves state reachability while eliminating unnecessary infinite regressions in backward search. The transformation subsumes the more usual arguments that apply only to strongly typed protocols.

Jorge R. Cuellar

Title: System security for the mobile Internet

Abstract: The IETF has now large amount of interesting, real-life, security protocols. It would be worthwhile that security protocol verification experts analyze the various proposals that are currently emerging, and detect errors or prove correctness at a very early stage of the design. This is possible since the IETF is open, the proposals are available online as Internet-drafts, and the discussion is done in open mailing lists. The talk gives an overview of some current mobility proposals, together with some preliminary formalization of the security goals.

Grit Denker

Title: Extending the CAPSL Integrated Protocol Environment for Multicast Protocols

Abstract: Protocols for secure group management are essential in a variety of application domains, such as confidential authenticated communication among coalition members, authenticated group decisions, or the secure administration of group membership and access control. The analysis of group management protocols poses new challenges on formal analysis methods. New language features and semantics are necessary to appropriately capture the concepts of such protocols. Analysis techniques and tools have to be revised and extended.

Multicast CAPSL (MuCAPSL) is an extension of the Common Authentication Protocol Specification Language (CAPSL) for secure multicast. MuCAPSL includes features such as a high-level organization of protocols into suites, a separation of roles within a protocol, group attributes to capture modifiable persistent state information of group members, and variable-length data structures such as arrays and sequences that are being used as fields in messages or state variables of agents.

Hand in hand with the extension of the language go extensions in the underlying semantic model that is based on a multiset rewriting approach. New "customized" predicates for group member state, protocol state, and multicast messages, as well as attacker memory and security goals are introduced.

We illustrate MuCAPSL and its semantics with the help of the Group Diffie-Hellman Protocol (GDH), an extension of the Diffie-Hellman Key Distribution to Group Communication. We have some preliminary experimentation results in using the Maude Model Checker to analyse GDH and uncovering security flaws.

Joint work with Jon Millen and Vitaly Shmatikov (SRI International)

Nancy Durgin

Title: A compositional logic for protocol correctness

Abstract: We present a specialized protocol logic that is built around a process language for describing the actions of a protocol. In general terms, the relation between logic and protocol is like the relation between assertions in Floyd-Hoare logic and standard imperative programs. Like Floyd-Hoare logic, our logic contains axioms and inference rules for each of the main protocol actions and proofs are protocol-directed, meaning that the outline of a proof of correctness follows the sequence of actions in the protocol. We prove that the protocol logic is sound, in a specific sense: each provable assertion about an action or sequence of actions holds in any run of the protocol, under attack, in which the given actions occur. This approach lets us prove properties of protocols that hold in all runs, while explicitly reasoning only about the sequence of actions needed to achieve this property. In particular, no explicit reasoning about the potential actions of an attacker is required.

Joint work with John Mitchell and Dusko Pavlovic

Michael Goldsmith

Title: The CSP approach

Abstract: Formal Systems' property checker FDR was at the heart of pioneering work in the application of state-exploration techniques to crypto-protocol analysis in the mid 1990s, as part of a series of strategic research projects of the UK Defence Research Agency. The tool has continued to develop, improvements in performance and scope being complemented by incorporating the conclusions of theoretical results established by conventional mathematics, while Lowe's Casper preprocessor has reduced the time needed to present a protocol for analysis from the man-months of modelling required during the development of the idiom to a matter of minutes.

But the simple Alice-and-Bob style of protocol with a Dolev-Yao attacker has become a well-trodden field, and other techniques such as backward symbolic search have generally overtaken FDR's performance. New challenges for the tool are sought, taking advantage

of its ability to deal with more than just trace properties, using refusal and stabilisation properties to model kinds of denial-of-service resiliance, for example.

Dieter Gollmann

Title: Authentication by correspondence

Abstract: The application of formal methods to security problems is often motivated by reciting the mantra that the design of security protocol s is difficult and error-prone. I disagree and claim that capturing security properties, formally or informally, is difficult and error-prone. For illustration, consider the history of the correspondence properties used to express authentication. In the early 1990s an "attack" was observed, which suggested that that two-way authentication is different from one-way authentication performed twice (in both directions). To find a property violated by this attack, matching computations were introduced in defining the requirements an authentication protocol should meet. Over a series of papers, this idea was further formalized and correspondence properties became the standard way of capturing authentication properties in formal analyses.

On closer inspection, one can see that the abovementioned attack has a meaningful explanation in a connection-oriented setting but does not make much sense in a connectionless communications system. Furthermore, strong correspondence requirements such as those used in the Bellare-Rogaway model require protocol executions to simulate connections. At the same time, there exist examples of protocols that have been formally verified in this very model but can be attacked when a protocol participant deviates from the protocol rules. These observations suggest that the origins of correspondence lie in a world that instinctively thought connection-oriented and ended up with capturing ideas related to connections rather than to verifying identities. Also, much of the early work on definitions and verification assumes that protocol participants are "honest". Abandoning this assumption forces us not only to re-examine our proofs, but also the definitions of protocol goals themselves.

Bart Jacobs

Title: JavaCard program verification

Abstract: This talk consists of two parts. The first one will describe the work on tool-assisted specification and verification for JavaCard that is being done within a recent European RTD project "VerifiCard" (see `http://www.verificard.org/`). JavaCard is a simplified version of Java, adapted for programming the latest generation of smart cards. Because these cards allow multiple applets and post-issuance downloading of applets, they involve considerable security risks, but also real challenges for the formal methods community.

The second part of the talk focusses on the contribution of Nijmegen within this project, namely JavaCard API and applet specification with JML (see `http://www.`

cs.iastate.edu/~leavens/JML.html), and verification with the prooftool PVS (see http://pvs.csl.sri.com/).

Peeter Laud

Title: Handling cryptographic primitives in an analysis for secure information flow.

Abstract: Given a program in a simple imperative programming language, and two subsets of program variables — "secret inputs" and "public outputs", we are interested, whether the final values of public outputs are independent of the initial values of secret inputs. The programming language contains encryption as a primitive operation, the cryptography is "real", i.e. the semantics of the encryption operator is a function working on bit-strings and satisfying certain complexity-theoretical assumptions.

Central to our approach is the abstraction of probability distributions over program states by sets of pairs of sets of program variables. The abstraction of a certain distribution collects all those pairs of sets of variables where the values of the variables of the first set are independent of the values of the variables of the second set.

The concrete semantics of the program transforms the distribution of program states at the beginning of the program to the distribution of program states at the end. The abstract semantics works at the abstractions of those distributions. It satisfies the following correctness criterion: if the values of the variables in a set X are not independent of the values of the variables in a set Y at the end of the program, then the pair (X, Y) does not belong to the application of the abstract semantics to the abstraction of the initial distribution.

The values of the variables in a set X are said to be independent of the values of the variables in a set Y for a certain distribution D over program states, if the result of sampling D once and picking the values of the variables in X and Y from that sample looks the same as the result of sampling D twice and picking the values of the variables in X from the first sample and the values of the variables in Y from the second.

Gavin Lowe

Title: Analysing protocols subject to password guessing attacks

Abstract: In this talk we consider guessing attacks upon security protocols, where an intruder guesses one of the values used (typically a poorly-chosen password) and then seeks to verify that guess. We formalise such attacks, and in particular the way in which the guess is verified. We then describe how to model such attacks within the process algebra CSP, so that they can be detected using the model checker FDR.

Catherine Meadows

Title: Experiences in the analysis of the GDOI protocol

Abstract: Although research in the application of formal methods to cryptographic protocol analysis has been growing rapidly, it has of yet had little influence on the design and implementation of protocols intended for actual use. This is not because the designers of cryptographic protocols do not recognize the important of assurance, but rather seems the result of the fact that currently there are no good pathways for introducing formal analysis into the design process. In this paper we describe how we are attempting to help remedy this lack by working with the MSec working group in the Internet Engineering Task Force on the design and analysis of the Group Domain of Interpretation Protocol (GDOI), a secure multicast protocol intended to work with the Internet Key Exchange protocol. The purpose of our work has been two-fold: first, to identify and correct errors and ambiguities early on, and secondly to speed up the standardization process by providing increased evidence of GDOI's soundness. We give a brief description of our ongoing work in the analysis of GDOI, and point both the benefits realized by the analysis and some of the open questions that raised by our experiences.

Joint work with Iliano Cervesato and Paul Syverson.

Title: A cost-based framework for analysis of denial of service in networks

Abstract: Denial of service is becoming a growing concern. As computer systems communicate more and more with others that they know less and less, they become increasingly vulnerable to hostile intruders who may take advantage of the very protocols intended for the establishment and authentication of communication to tie up resources and disable servers. We show how some principles that have already been used to make cryptographic protocols more resistant to denial of service by trading off the cost to defender against the cost to the attacker can be formalized based on a modification of the Gong-Syverson fail-stop model of cryptographic protocols, and indicate the ways in which existing cryptographic protocol analysis tools could be modified to operate within this formal framework. We also indicate how this framework could be extended to protocols that do not make use of strong authentication.

Title: Environmental requirements and authentication

Abstract: Most work on requirements in the area of authentication protocols has concentrated on identifying requirements for the protocol without much consideration of context. Little work has concentrated on assumptions about the environment, for example, the applications that make use of authenticated keys. We showhow the interaction between a protocol and its environment can have a major effect on a protocol. Specifically we demonstrate a number of attacks on published and/or widely used protocols that are not feasible against the protocol running in isolation (even with multiple runs) but become feasible in some application environments. We also discuss the tradeoff between putting constraints on a protocol and putting constraints on the environment in which it operates.

Joint work with Ran Canetti and Paul Syverson.

Jon Millen

Title: Constraint solving

Abstract: Security for cryptographic protocols can be expressed as a reachability problem, which is decidable for a finite number of processes. Reachability is transformed to a constraint-set satisfiability problem, which is solved using transformation rules based on attacker term-construction operations. The message model is a free term algebra and permits non-atomic keys.

Lawrence Paulson

Title: Verification of SET: The purchase phase

Abstract: Past work on protocol verification has largely focused on simple protocols from the academic world. SET is a huge protocol devised by Visa and Mastercard for Internet shopping. It aims to protect both cardholders and merchants from fraud. Protocol participants must first register with their bank, which (after making suitable checks) will provide them with electronic credentials. Customers don't give their credit card numbers directly, but instead give these credentials to the merchant to prove their honesty. The merchant presents similar credentials to the customer. For payment, the customer's account details are passed to the merchant's bank, but not to the merchant himself.

The initial registration phase could in principle be simple. Unfortunately, complex mechanisms (e.g. digital envelopes) and unnecessary encryption complicate the proofs. The talk gives a very high-level overview of the SET protocol and then shows a few details of the proofs of its registration and payment phases.

Andreas Pfitzmann

Title: A security guy's remarks to the formal methods community

Abstract: Confidentiality properties may be harder to prove than integrity and availability properties. A possibilistic framework is not enough to prove security (confidentiality in particular), but only enables to find some flaws. To prove security, a probabilistic framework is needed.

Security properties which are not achieved perfectly may be harder to prove, but these non-perfect security properties are often the best you can achieve in practice. Then, counting arguments of equally probable possibilities are not enough.

Very large and varying numbers of participants may participate in future protocols. In addition, the security property to be proved may depend on the number of participants and even the development of their number. Security protocols with that property are nearly all protocols providing anonymity properties in general and anonymous communication in particular.

Future protocol runs may be long lived in the sense that in the midst of the protocol run, some participants may get dishonest (e.g. their Windows ME machine gets subverted), but others get honest (e.g. they re-install their Windows operating system and application software and disable ActiveX).

Tools to prove properties of cryptographic protocols should be used or even enhanced to help in the transformation of security protocols or even their design. As an optimizing compiler may detect dead code and omit it, proof systems should tell the protocol designer that by omitting this and that operation of her/his protocol, the security property may nevertheless be proved (or nevertheless no counter example can be found). This may not only help to arrive at leaner security protocols, but also to get smaller proofs giving more insight.

It is not enough to specify all security properties a protocol is intended to achieve, but in addition the complete environment, in particular all security relevant assumptions, have to be specified. If this is not done, you neither can hope that security can be proved nor that security protocols are compositional. An example what happens if relevant properties of the environment are ignored is a "secure" remote login protocol (SSH), which transmits each keystroke of the human user in a separate message immediately: Even when all crypto is perfect (probabilistic encryption), the mere timing of the messages sent gives an attacker information he can exploit - the time intervals between messages correspond to the time intervals between keystrokes - and the time intervals give, when observed long enough and analyzed cleverly, all characters typed with significant probability. That way, an attacker gets valuable information on all passwords and on all pass phrases, e.g. to unlock cryptographic keys.

Michaël Rusinowitch

Title: Deciding insecurity for non-atomic Keys

Abstract: We present an NP decision procedure for insecurity in the case of finite sessions and non-atomic encryption keys (and no fixed bound on the size of messages). We also explain that in order to build an attack with a fixed number of sessions the intruder needs only to forge messages of linear size, provided that they are represented as dags.

This is joint work with Mathieu Turuani and is based on our paper presented in June 2001 at 14th CSFW. Further information available at: `http://www.loria.fr/equipes/protheo/SOFTWARES/CASROL/`.

André Scedrov

Title: Inductive methods and contract-signing protocols

Abstract: Distributed contract signing over a network involves many challenges in mimicking the features of paper contract signing. For instance, a paper contract is usually signed by both parties at the same time and at the same place, but distributed electronic

transactions over a network is inherently asymmetric in that someone has to send the first message.

Several digital contract-signing protocols have been devised in order to overcome this basic asymmetry and to achieve symmetric properties such as fairness, namely: 1) if nothing goes wrong, both participants receive a valid contract, 2) every participant may complete the protocol, and 3) either both participants receive a valid contract or neither one does. Such protocols often involve a trusted third party that can enforce the contract after it witnesses a partial completion of the protocol. In optimistic contract-signing protocols the trusted third party is contacted only in case of a dispute, otherwise the protocol can be completed without involving the third party. Such protocols involve several subprotocols that allow a contract to be signed normally or aborted or resolved by the trusted third party.

Another kind of symmetry desirable in distributed contract signing was identified by Garay, Jakobsson, and MacKenzie: a fair protocol is said to be abuse-free if, at any stage of the protocol, it is impossible for any participant, say A, to be able to prove to an outside challenger that A has the power to choose between completing the contract and aborting it. A formal analysis of the Garay-Jakobsson-MacKenzie two-party contract-signing protocol was carried out by Mitchell and Shmatikov using a finite state verification tool. They showed that negligence of the trusted third party may lead to loss of abuse-freeness or fairness. Based on their analysis, Mitchell and Shmatikov also suggested a revision of the protocol. This revised version is our reference point.

We study an approximation of the abuse-freeness property, namely, at any stage of a fair protocol, any protocol participant does not have both the power to complete the contract as well as the power to abort it. As noted by Mitchell and Shmatikov, this property is not trace-based. We use a multiset-rewriting formalism for protocol analysis to formally state this property in terms of a certain recursive property of the protocol execution tree, which we then prove by inductive methods. Our proof relies on a strong notion of fairness adopted from which itself we formally state in the multiset-rewriting formalism and prove by inductive methods. We also show that our approximation of abuse-freeness may be represented in terms of provability in a logical system, in which formal derivations correspond to full execution trees and vice versa.

This work was carried out in collaboration with Rohit Chadha and Max Kanovich.

Vitaly Shmatikov

Title: Defining anonymity

Abstract: We consider several notions of anonymity and privacy and develop uniform, formal definitions of what it means for a communication protocol to be "anonymous." The definitions are based on observational invariance of protocols under restricted substitutions. Defining anonymity in this way clarifies the relation between concepts such as sender and recipient anonymity, untraceability, unlinkability, and privacy, and enables formal verification of anonymity properties.

Joint work with Dominic Hughes.

Scott D. Stoller

Title: Generation of environments for distributed systems

Abstract: Testing and verification of open distributed systems require a model of the system's environment. For example, a web server can be regarded as a system that interacts with an environment containing browsers, possibly buggy. A distributed system with security requirements can be regarded as operating in the environment of an adversary-controlled network. It is often useful to have a model of a system's most general environment, i.e., one that exercises all possible behaviors of the system. We describe a static analysis for Java that computes a partition of the system's inputs: inputs in the same equivalence class lead to identical behavior. The partition provides a basis for generation of code for a most general environment.

Irfan Zakiuddin

Title: Making key distribution robust, or an exercise in deploying threshold cryptography

Abstract: Threshold cryptography enables us to make key management services, such as session key distribution, robust by secure replication. However, we need sound protocols to achieve session key distribution that is secure and tolerant to failure or compromise of servers. In this work we use threshold cryptographic techniques from the open literature as a basis for formally specifying protocols to initialise servers and to issue session keys. Both protocols are distributed and tolerant to: network fault, server failure or server compromise. Formal specification elucidates the complex messaging that implements these services; it specifies the functionality that the services depend on, and it enables formal verification and validation.

Joint work with Bill Roscoe and Paul Gardiner

4 Followup activities

Participants expressed interest in archiving the slides of presentations. There is now a webpage, `http://www.informatik.uni-freiburg.de/~accorsi/dagstuhl/`, that contains this information. Moreover, current plans are for producing a "state-of-the-art" based, in part, on the workshop. Future plans also include the design of web-site providing a centralized bibliography service on verification of security protocols. This web-site, which is currently in experimental phase, can be found at `http://www.informatik.uni-freiburg.de/~accorsi/protsecweb/`.