Report from Dagstuhl Seminar 12482

Analysis of Security APIs

Edited by
Mike Bond[1], Riccardo Focardi[2], Sibylle Fröschle[3], and Graham Steel[4]

1 **University of Cambridge, GB,** mike.bond@cl.cam.ac.uk
2 **Università Ca' Foscari di Venezia, IT,** focardi@dsi.unive.it
3 **Universität Oldenburg, DE**
4 **INRIA, FR,** graham.steel@inria.fr

Abstract

This report documents the programme and the outcomes of Dagstuhl Seminar 12482 "Analysis of Security APIs". Abstracts from the talks give a snapshot of current research in the field, while reports on the discussions give a roadmap for future research in the area.

Seminar 26.–28. November, 2012 – www.dagstuhl.de/12482
1998 ACM Subject Classification K.6.m Miscellaneous, Security
Keywords and phrases Security APIs, cryptography, key management, formal methods, security protocols
Digital Object Identifier 10.4230/DagRep.2.11.155

1 Executive Summary

Mike Bond
Riccardo Focardi
Sibylle Fröschle
Graham Steel

This report documents the programme and outcomes of Dagstuhl Seminar 12482 "Analysis of Security APIs". The seminar brought together 32 participants from academia and industry in Europe and the USA. It featured a joint session with the concurrent seminar on quantitative security analysis (which included the keynote talk), a breakout session with demonstrations of software and practical classes, a discussion of the most important open problems in the field and a collection of talks spanning the breadth of the field from theoretical models to applications.

Research Context and Goals of the Seminar

A security API is an Application Program Interface that allows untrusted code to access sensitive resources in a secure way. It is the interface between processes running with different levels of trust. Examples of security APIs include the interface between the tamper-resistant chip on a smartcard (trusted) and the code running on the client application (untrusted), the interface between a cryptographic Hardware Security Module (or HSM, trusted) and the host machine (untrusted), and web service APIs (an interface between a server, trusted by the service provides, and the rest of the Internet).

The crucial aspect of a security API is that it is designed to enforce a policy, i.e. no matter what sequence of commands in the interface are called, and no matter what the parameters, certain security properties should continue to hold. This means that if the less trusted code turns out to be malicious (or just faulty), the carefully designed API should prevent compromise of critical data. Designing such an interface is extremely tricky and error prone, and over the last ten years, serious vulnerabilities in the security APIs deployed in HSMs in the ATM (cash machine) network and in commodity security devices like smartcards and USB keys have come to light.

A number of formal methods researchers have turned their attention to security APIs over the last five years. While significant advances have been made and notable results achieved, such as the discovery of several new attacks, the process of specifying and verifying the security policy for such APIs still lacks both satisfactory foundations and efficient algorithms. At the same time, the security API paradigm is being proposed as a solution for more and more applications, from social networks to smartphones, with more complicated and less well understood security goals.

The objective of the seminar was to bring together researchers in academia and industry around the topic of security APIs and their analysis. There were three main aims:

1. To address the shortcomings of current API analysis techniques as applied to the relatively well explored domains of cryptographic key management and HSMs, in particular in their ability to deal with global mutable state and their models of cryptography.
2. To identify the security API requirements arising from the new generation of applications, in mobile device applications and web services, and map out the research problems that need to be solved in order that formal API analysis can be applied here.
3. To find ways to include the process and results of formal API analysis into the standards and certification procedures.

Some progress was made on all these points in the talks and the discussions late into the evening that followed in the conducive environment of Schloss Dagstuhl. On the first point, several talks presented models specifically aimed at modelling state in a more satisfactory way, and we had a tutorial on the verification methods used in program analysis. Several new application areas for API analysis were presented, including car to car communication and password protection. Some highly enlightening talks on the standards process helped to improve understanding of the problem, if not providing steps to an easy solution. The variety of open problems identified (see summary below) shows that this is a vibrant area of computer security research with much promise for the future.

2 Table of Contents

Executive Summary
Mike Bond, Riccardo Focardi, Sibylle Fröschle, Graham Steel 155

Keynote: Ross Anderson – Security evolution: interaction of economics and APIs 159

Overview of Talks 159

Security in Car2X Communication
Daniel Angermeier 159

Demo of Tookan: Tool for Cryptoki Analysis
Romain Bardou 160

Tokenisation – pseudo-security and compliance engineering
Mike Bond 160

Revoke and Let Live: A Secure Key Revocation API for Cryptographic Devices
Veronique Cortier 161

Hands-on tutorial on Padding Oracle Attacks
Riccardo Focardi 161

Challenges in Security API Verification
Sibylle Froeschle 162

Universally Composable Key-Management
Steve Kremer 162

Formal Security Analysis Results for the Yubikey and YubiHSM
Robert Kuennemann 162

A Framework for the Cryptographic Verification of Java-like Programs
Ralf Kuesters 163

Lazy Mobile Intruders
Sebastian Moedersheim 164

Temporal Information Flow
Markus N. Rabe 164

APIs and Cryptography
Phillip Rogaway 165

Automated Reasoning on Data Structures
Viorica Sofronie-Stokkermans 165

MAC in the Box
Graham Steel 165

Verification of a Trusted Virtual Security Module
Ronald Toegl 166

Discussion 166

What I learned 166

Future research topics 166

Conclusion 167

Participants 168

3 Keynote: Ross Anderson – Security evolution: interaction of economics and APIs

Ross Anderson opened the second day of the seminar in a joint session with the quantitative security analysis workshop[1]. The talk was an exciting journey through the evolution of IT security from both technical and socio-economic perspectives. Economics matters: failures are inversely proportional to how much is invested into guarding and fixing a system; lack of security is often considered an external, independent factor we might need to address, just like environmental pollution. Moreover, security is often an obstacle: to win the market a new platform has little security so that development is easier; once the market is captured security can be faced. Ross finally illustrated a few impressive "non-success stories" on payment and banking systems and pointed out how the APIs often are the places of interest, i.e. "where the rubber hits the road".

4 Overview of Talks

The technical programme included demos, hand-on sessions, perspectives, work in progress and new ideas as well as conventional research talks.

4.1 Security in Car2X Communication

Daniel Angermeier (Fraunhofer AISEC – München, DE)

Joint work of Angermeier, Daniel; C2C Communication Consortium

Car2X communication promises improvements in modern traffic, as cooperative driving might help avoid dangerous situations. Furthermore, C2X communication aims to achieve improved traffic efficiency. However, these potential advantages are opposed by risks caused especially by attacks on Intelligent Transport Systems (ITS) trying to abuse these new features. Therefore, security plays a major role in ITSs to reach the afore mentioned goals and to avert threats caused by attackers. In my talk, I will highlight a few aspects of security in C2X communication, which focus on fulfilling the special requirements in C2X communication, like e.g., privacy of ITS users or proof of trustworthiness of received messages.

[1] http://www.dagstuhl.de/12481

4.2 Demo of Tookan: Tool for Cryptoki Analysis

Romain Bardou (INRIA – Paris, FR)

Joint work of Bardou, Romain; Focardi, Riccardo; Steel, Graham
Main reference M. Bortolozzo, M. Centenaro, R. Focardi, G. Steel, "Attacking and Fixing PKCS#11 Security Tokens," in Proc. of the 17th ACM Conf. on Computer and Communications Security (CCS'10), pp. 260–269, ACM, 2010.
URL http://dx.doi.org/10.1145/1866307.1866337
URL http://tookan.gforge.inria.fr/

This is a demonstration of Tookan, a tool which automatically finds attacks on cryptographic devices. Tookan reverse engineers the behavior of PKCS#11 tokens. It learns the preconditions of each command to build a model of the token. It then runs a model-checking analysis on the model to try and find a sequence of commands leading to an invalid state. Tookan can be used for penetration testing, but it can also be used to compare device configurations, or to help develop a safe cryptographic device.

4.3 Tokenisation – pseudo-security and compliance engineering

Mike Bond (University of Cambridge, GB)

This talk describes the challenges in designing a security solution whose main goal is to satisfy compliance requirements for international financial standards such as the PCI Data Security Standard (PCI DSS). Together with highly prescriptive internal security standards there is often very little room to design an elegant or efficient solution and this proves very costly for organisations that must be compliant. Sometimes the compliance rules even fly in the face of security concerns, or are contradictory/ill-defined. For instance, tokenisation is defined by some to be a deterministic substitution mechanism which is not an algorithmic function, yet a look-up table is indeed a function. Sometimes the challenge becomes to modify a solution design to avoid falling foul of compliance rules without introducing significant vulnerability, and sometimes the challenge is to actively frustrate trivial data flow analysis of a solution such as is used by many auditors who simply follow the flows of keys and data and then make broad prescriptions about the wisdom or otherwise of a scheme. The talk proposes some schemes for 'audit-resistant cryptography', and shows their application to practical problem solving in an environment tainted by conflicting security and compliance requirements.

4.4 Revoke and Let Live: A Secure Key Revocation API for Cryptographic Devices

Veronique Cortier (CNRS – Nancy, FR)

Main reference V. Cortier, G. Steel, C. Wiedling, "Revoke and Let Live: A Secure Key Revocation API for Cryptographic Devices," in Proc. of the 19th ACM Conf. on Computer and Communications Security (CCS'12), pp. 918–928, ACM, 2012.
URL http://dx.doi.org/10.1145/2382196.2382293

While extensive research addresses the problem of establishing session keys through cryptographic protocols, relatively little work has appeared addressing the problem of revocation and update of long term keys. We present an API for symmetric key management on embedded devices that supports revocation and prove security properties design in the symbolic model of cryptography. Our API supports two modes of revocation: a passive mode where keys have an expiration time, and an active mode where revocation messages are sent to devices. For the first we show that once enough time has elapsed after the compromise of a key, the system returns to a secure state, i.e. the API is robust against attempts by the attacker to use a compromised key to compromise other keys or keep the compromised key alive past its validity time. For the second we show that once revocation messages have been received the system is immediately in a secure state. Notable features of our designs are that all secret values on the device are revocable, and the device returns to a functionally equivalent state after revocation is complete.

4.5 Hands-on tutorial on Padding Oracle Attacks

Riccardo Focardi (Università Ca' Foscari di Venezia, IT)

Joint work of Bardou, Romain; Focardi, Riccardo; Kawamoto, Yusuke; Simionato, Lorenzo; Steel, Graham; Tsay, Joe-Kai
Main reference R. Bardou, R. Focardi, Y. Kawamoto, L. Simionato, G. Steel, J. Tsay, "Efficient Padding Oracle Attacks on Cryptographic Hardware," in Proc. of 32nd Annual Conf. on Advances in Cryptology (CRYPTO'12), LNCS, Vol. 7417, pp. 608–625, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-32009-5_36
URL http://secgroup.ext.dsi.unive.it/wp-content/uploads/2012/11/Practical-Padding-Oracle-Attacks-on-RSA.html

We revise attacks on the RSA cipher based on side-channels that leak partial information about the plaintext. We show how to compute a plaintext when only its parity is leaked. We then describe PKCS#1 v1.5 padding for RSA and we show that the simple leakage of padding errors is enough to recover the whole plaintext, even when it is unpadded or padded under another scheme. This vulnerability is well-known since 1998 but the flawed PKCS#1 v1.5 padding is still broadly in use. We discuss recent optimizations of this padding oracle attack that make it effective on commercially available cryptographic devices. We illustrate through many examples and fragments of code. This tutorial is based on the paper appeared in Hakin9 – Defend Yourself! Hands-on Cryptography, September 2012, available at http://secgroup.ext.dsi.unive.it/wp-content/uploads/2012/11/Practical-Padding-Oracle-Attacks-on-RSA.html

4.6 Challenges in Security API Verification

Sibylle Froeschle (OFFIS – Oldenburg, DE)

Main reference S. Froeschle, "Causality in security protocols and security APIs: foundations and practical verification," Habilitation Thesis, University of Oldenburg, 2012.
URL https://vhome.offis.de/sibyllef/habil.pdf

In this talk we pinpoint key challenges in security API verification and suggest possible solutions and research directions. Among the challenges we discuss are the problem of scale and several general aspects such as how to specify security APIs and what to verify about them. A central theme will be how to deal with the key metadata that governs how a key entity is managed by the API. The talk is based on a comparative study of PKCS#11 and IBM's CCA, two widely deployed key management APIs. A detailed discussion can be found in Chapter I.4 of [1]

References

1 Sibylle Fröschle. Causality in security protocols and security APIs: foundations and practical verification. Habilitation thesis, University of Oldenburg, 2012.

4.7 Universally Composable Key-Management

Steve Kremer (INRIA Grand Est – Nancy, FR)

Main reference St. Kremer, R. Künnemann, G. Steel, "Universally Composable Key-Management," Cryptology ePrint Archive: Report 2012/189, 2012.
URL http://eprint.iacr.org/2012/189

We present the first universally composable key management functionality, formalized in the GNUC framework by Hofheinz and Shoup. It allows the enforcement of a wide range of security policies and can be extended by diverse key usage operations with no need to repeat the security proof. We illustrate its use by proving an implementation of a security token secure with respect to arbitrary key-usage operations and explore a proof technique that allows the storage of cryptographic keys externally, a novel development in simulation-based security frameworks.

4.8 Formal Security Analysis Results for the Yubikey and YubiHSM

Robert Kuennemann (CNRS, INRIA, FR)

The Yubikey is a small hardware device designed to authenticate a user against network-based services. Despite its widespread adoption (over a million devices have been shipped by Yubico to more than 20 000 customers including Google and Microsoft), the Yubikey protocols have received relatively little security analysis in the academic literature. In the first part of this paper, we give a formal model for the operation of the Yubikeyone-time password (OTP) protocol. We prove security properties of the protocol for an unbounded number of fresh

OTPs using a protocol analysis tool, tamarin. In the second part of the talk, we analyze the security of the protocol with respect to an adversary that has temporary access to the authentication server. To address this scenario, Yubico offers a small Hardware Security Module (HSM) called the YubiHSM, intended to protect keys even in the event of server compromise. We show if the same YubiHSM configuration is used both to set up Yubikeys and run the authentication protocol, then there is inevitably an attack that leaks all of the keys to the attacker. Our discovery of this attack lead to a Yubico security advisory in February 2012. For the case where separate servers are used for the two tasks, we give a configuration for which we can show using the same verification tool that if an adversary that can compromise the server running the Yubikey-protocol, but not the server used to set up new Yubikeys, then he cannot obtain the keys used to produce one-time passwords.

4.9 A Framework for the Cryptographic Verification of Java-like Programs

Ralf Kuesters (Universität Trier, DE)

Joint work of Küsters, Ralf; Truderung, Tomasz; Graf, Jürgen
Main reference R. Küsters, T. Truderung, J. Graf, "A Framework for the Cryptographic Verification of Java-like Programs," in Proc. of the 25th IEEE Computer Security Foundations Symp. (CSF'12), pp. 198–212, IEEE CS, 2012.
URL http://dx.doi.org/10.1109/CSF.2012.9

We consider the problem of establishing cryptographic guarantees—in particular, computational indistinguishability—for Java or Java-like programs that use cryptography. For this purpose, we propose a general framework that enables existing program analysis tools that can check (standard) non-interference properties of Java programs to establish cryptographic security guarantees, even if the tools a priori cannot deal with cryptography. The approach that we take is new and combines techniques from program analysis and simulation-based security. Our framework is stated and proved for a Java-like language that comprises a rich fragment of Java. The general idea of our approach should, however, be applicable also to other practical programming languages. As a proof of concept, we use an automatic program analysis tool for checking non-interference properties of Java programs, namely the tool Joana, in order to establish computational indistinguishability for a Java program that involves clients sending encrypted messages over a network, controlled by an active adversary, to a server. The approach may also be applicable for checking security properties of Java programs that use security APIs.

4.10 Lazy Mobile Intruders

Sebastian Moedersheim (Technical University of Denmark, DK)

Joint work of Moedersheim, Sebastian; Nielson, Hanne Riis; Nielson, Flemming
Main reference S. Moedersheim, F. Nielson, H.R. Nielson, "Lazy Mobile Intruders," in Proc. of the 2nd Int'l Conf. on Principles of Security and Trust (POST'13), LNCS, Vol. 7796, pp. 147–166, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-36830-1_8
URL http://www.imm.dtu.dk/ samo/mobile.pdf

We present a new technique for analyzing platforms that execute potentially malicious code, such as web-browsers, mobile phones, or virtualized infrastructures. Rather than analyzing given code, we ask what code an intruder could create to break a security goal of the platform. To avoid searching the infinite space of programs that the intruder could come up with (given some initial knowledge) we adapt the lazy intruder technique from protocol verification: the code is initially just a process variable that is getting instantiated in a demand-driven way during its execution. We also take into account that by communication, the malicious code can learn new information that it can use in subsequent operations, or that we may have several pieces of malicious code that can exchange information if they "meet". To formalize both the platform and the malicious code we use the mobile ambient calculus, since it provides a small, abstract formalism that models the essence of mobile code.

References

1 Sebastian Mödersheim, Flemming Nielson, Hanne Riis Nielson. *Lazy Mobile Intruders*. In Proceedings of POST 2013, Springer LNCS, 2013. Extended version available as IMM-TR-2012-13 at www.imm.dtu.dk/~samo.

4.11 Temporal Information Flow

Markus N. Rabe (Universität des Saarlandes, DE)

Joint work of Dimitrova, Rayna; Finkbeiner, Bernd; Kovács, Máté; Rabe, Markus N.; Seidl, Helmut
Main reference R. Dimitrova, B. Finkbeiner, M. Kovács, M.N. Rabe, H. Seidl, "Model Checking Information Flow in Reactive Systems," in Proc. of the 13th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI'12), LNCS, Vol. 7148, pp. 169–185, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-27940-9_12
URL http://www.react.uni-saarland.de/publications/DFKRS12.html

There is a great number of different security properties that are discussed in the various security communities, but they are defined on different semantic models and are thus difficult to compare. Further, there is currently little interface to other specifications that concern the safety and lifeness aspects of a system. I will present recent results on how to integrate secrecy properties into temporal logics by introducing a new modal operator. Besides providing a common framework for many security properties, this allows to precisely specify when and under which conditions a variable has to be kept secret and also until when the secrecy needs to be maintained. I also give an overview of the complexity results and an efficient fragment that is suitable for checking large models.

4.12 APIs and Cryptography

Phillip Rogaway (University of California – Davis, US)

In this talk I discussed, in turn: (1) how an API can inform a cryptographic definition (example: the notion of online indistinguishability from [Rogaway, Wooding, Zhang 2012]); (2) how an API can inform the design of a cryptographic algorithm (example: incremental encryption and OCB3 [Krovetz, Rogaway 2011]); and (3) how cryptographic expertise can (maybe poorly) inform the design of an API (example the GCS-API [Rogaway 1994] that I developed at IBM).

4.13 Automated Reasoning on Data Structures

Viorica Sofronie-Stokkermans (Universität Koblenz-Landau, DE)

We present a class of theories for which the problem of checking satisfiability of ground formulae is decidable. Examples are theories of data structures (fragments of theories of arrays or pointers), as well as extensions of certain classes of base theories with functions which satisfy certain recursion and homomorphism properties. We present the applications of these ideas in verification and possibly also in cryptography.

4.14 MAC in the Box

Graham Steel (CNRS, INRIA, FR)

We propose to construct a formally verified open source security device for calculating messages authentication codes (MACs). The application we have in mind is the storage of user passwords on a web server. Typically, these are stored as salted hashes of the password and some other diversifiers (such as username). Unfortunately, password files are often leaked after server compromise, and computing power is sufficiently affordable to allow brute force offline cracking of the passwords. To prevent this we propose to calculate a keyed hash (HMAC) of passwords. All HMACs will be calculated in a separate hardware device (the MAC in the Box, or MITB) where the key is stored. The API of the device will allow calculation and verification of HMACs but no commands will give access to the key. In the event of server compromise, the attacker's ability to crack passwords is limited by the throughput of the MITB. After the compromise is discovered and the attacker is ejected, the password file is of no use to him, since he has no access to the HMAC key. We anticipate that such a simple device could be formally verified to a low level. In combination with a low cost and open source design the MITB will be an attractive best-practice option for website administrators.

4.15 Verification of a Trusted Virtual Security Module

Ronald Toegl (TU Graz, AT)

Joint work of Toegl, Ronald; Reimair, Florian; Pirker, Martin
Main reference R. Toegl, F. Reimair, M. Pirker, "Waltzing the Bear, or: A Trusted Virtual Security Module," in Proc. of the 9th European PKI Workshop: Research and Applications (EUROPKI 2012), LNCS, Springer, in press.
URL https://online.tugraz.at/tug_online/voe_main2.getVollText?pDocumentNr=286273&pCurrPk=67562

Cryptographic key material needs to be protected. Currently, this is achieved by either pure software based solutions or by more expensive dedicated hardware security modules. We present a practical architecture to project the security provided by the Trusted Platform Module and Intel Trusted eXecution Technology on a virtual security module. Our approach uses commodity personal computer hardware to offer integrity protection and strong isolation to a security module which implements a compact security API that has been fully verified. Performance results suggest that our approach offers an attractive balance between speed, security and cost.

5 Discussion

In the final session, participants were asked to describe one thing they had learnt during the seminar and one important topic for future research (either to be conducted by themselves or by others). Here we highlight some of the most interesting suggestions:

5.1 What I learned

- The auditor as an adversary. Mike Bond's talk on the socio-technical side of standards and certification prompted several comments. It is clear that what formal researchers analyse is not always relevant for practice – e.g. because of standards. Also one can see that compliance is sometimes damaging usability and security. In particular, "certification as compliance" is seen as eroding to the value of the certification process.
- Even small APIs are useful and present interesting design and verification challenges
- Formal researchers were frequently struck by the range of applications of HSMs, and the practicalities of their use (e.g. in a mixed estate of heterogeneous configurations).
- Being an area that attracts theoreticians but also practitioners, many researchers found it interesting to consider the security economics angle of API analysis, as outlined by Ross Anderson in his talk.
- Practitioners were generally pleased to see that tools were on the way. In particular, competition between approaches seems healthy. There is no need to work on just one approach for the moment.
- Many attendees found the tension between theory and practice interesting

5.2 Future research topics

Many new security APIs ripe for analysis were suggested, including:
- Low level APIs of crypto devices
- OS (e.g. SE linux security modules).

- Geographic security APIs – e.g. vehicular.
- Heterogeneous networks of APIs – e.g. different HSMs with different APIs in networks.

Other topics included:

- Improved design of APIs around Crypto considerations
- Languages for human interpretation of APIs and policy. Human behavioural studies that could lead to comprehensible security policies.
- "Better than Dolev Yao" models (i.e. more cryptographic detail)
- A formally verified, open source HSM
- Concurrency and its effects on security – inside devices/drivers/applications
- Privacy properties of security APIs – e.g. in V2X
- A simple common language for APIs.

6 Conclusion

The field of security API analysis is in rude health. The seminar was over subscribed and the participation by attendees enthusiastic. As well as consolidating well known subjects in the area, the seminar identified new research directions in foundations and applications. The next few years should be an exciting time for research in this area.

Participants

- Pedro Adao
IST – TU of Lisbon, PT
- Ross Anderson
University of Cambridge, GB
- Daniel Angermeier
Fraunhofer AISEC – München, DE
- David R. Aspinall
University of Edinburgh, GB
- Romain Bardou
INRIA – Paris, FR
- Mike Bond
University of Cambridge, GB
- Veronique Cortier
CNRS – Nancy, FR
- Marion Daubignard
Direction Générale de l'Armement, FR
- Stéphanie Delaune
CNRS, ENS – Cachan, FR
- Riccardo Focardi
Univ. Ca' Foscari di Venezia, IT
- Sibylle Fröschle
OFFIS – Oldenburg, DE
- Steve Kremer
INRIA Grand Est – Nancy, FR
- Robert Künnemann
CNRS, ENS – Cachan, FR
- Ralf Küsters
Universität Trier, DE
- Flaminia L. Luccio
Univ. Ca' Foscari Venezia, IT
- Matteo Maffei
Universität des Saarlandes, DE
- Sebastian Mödersheim
Technical Univ. of Denmark, DK
- Benjamin Morin
ANSSI -Paris, FR
- Andreas Philipp
Utimaco Safeware AG, DE
- Markus N. Rabe
Universität des Saarlandes, DE
- Phillip Rogaway
Univ. of California – Davis, US
- Mark D. Ryan
University of Birmingham, GB
- Stefanie Schlegel
OFFIS – Oldenburg, DE
- Jörg-Cornelius Schneider
Deutsche Bank – Eschborn, DE
- Laurent Simon
University of Cambridge, GB
- Viorica Sofronie-Stokkermans
Universität Koblenz-Landau; MPI für Informatik, Saarbrücken
- Marco Squarcina
Univ. Ca' Foscari di Venezia, IT
- Graham Steel
CNRS, ENS – Cachan, FR
- Petr Svenda
Masaryk University, CZ
- Susan Thompson
MasterCard Worldwide, Warrington, GB
- Frank Thunig
Utimaco Safeware AG, DE
- Ronald Toegl
TU Graz, AT