Lecture Notes in Computer Science 8208

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Hanne Riis Nielson Dieter Gollmann (Eds.)

Secure IT Systems

18th Nordic Conference, NordSec 2013
Ilulissat, Greenland, October 18-21, 2013
Proceedings

Springer

Volume Editors

Hanne Riis Nielson
Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, Building 322, 2800 Lyngby, Denmark
E-mail: riis@imm.dtu.dk

Dieter Gollmann
Technische Universität Hamburg-Harburg
Institut für Sicherheit in verteilten Anwendungen / E-15
Harburger Schloßstraße 20, 21079 Hamburg, Germany
E-mail: diego@tuhh.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41487-9 e-ISBN 978-3-642-41488-6
DOI 10.1007/978-3-642-41488-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013950088

CR Subject Classification (1998): K.6.5, D.4.6, E.3, C.2, K.4.4, F.2, D.2, H.2.7

LNCS Sublibrary: SL 4 – Security and Cryptology

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

NordSec was initially started as a workshop series with the aim of bringing together researchers and practitioners working on computer security in the Nordic countries in 1996, thereby establishing a forum for discussion and cooperation between universities, industry, and computer societies. Since then, the workshop has developed into a fully fledged international conference, held in the Nordic (and Baltic) countries – with five events in Sweden, three in Norway and Finland, and two in Denmark, Iceland, and Estonia.

The 18th Nordic Conference on Secure IT Systems took place in Ilulissat (Jakobshavn) on Greenland during October 18–21, 2013. It had for a long time been a dream to arrange the conference in this remote part of the Danish Kingdom and the venue of Ilulissat was indeed remarkable – situated 200 km north of the Arctic Circle and neighboring UNESCO's World Heritage Centre of Ilulissat Icefjord.

NordSec addresses a broad range of topics within IT security and in 2013 the conference had a special focus on the security challenges of cyber-physical systems. A total of 35 submissions were received, each of which was reviewed by three Program Committee members. After a thorough discussion phase, 18 of the submitted papers were accepted as regular papers and three as short papers; they all appear in these proceedings. Additionally, the conference featured two invited talks, one by David Basin on "Developing Security Protocols by Refinement" and another by Gilles Barthe on "Towards Verified Implementations of Cryptographic Constructions."

We wish to thank all the people who invested time and energy to make NordSec 2013 a success: First and foremost come all the authors who submitted papers to NordSec and presented them at the conference. The members of the Program Committee together with the external reviewers worked hard in evaluating the submissions and, in some cases, to shepherd promising work. We would also like to thank Roberto Vigo, Alessandro Bruni, and Nataliya Skrypnyuk for assisting with local arrangements. Last but not least, special thanks goes to Karin Jensen at Greenland Travel for very competent assistance with travel arrangements.

The conference was sponsored by MT-LAB, a VKR Centre of Excellence for the Modelling of Information Technology (`www.MT-LAB.dk`).

August 2013 Qujanarsuaq

Dieter Gollmann
Hanne Riis Nielson

Organization

Conference and Program Chairs

Dieter Gollmann	Hamburg University of Technology, Germany
Hanne Riis Nielson	Technical University of Denmark, Denmark

International Program Committee

Tuomas Aura	Aalto University, Finland
Bengt Carlsson	Blekinge University of Technology, Sweden
Mads Dam	Royal Institute of Technology, Sweden
Nicola Dragoni	Technical University of Denmark
Rene Rydhof Hansen	Aalborg University, Denmark
Simone Fischer-Hübner	Karlstad University, Sweden
Chris Hankin	Imperial College London, UK
Erland Jonsson	Chalmers University of Technology, Sweden
Frank Kargl	University of Ulm, Germany
Svein Johan Knapskog	Norwegian University of Science and Technology
Hanno Langweg	Høgskolen i Gjøvik, Norway
Peeter Laud	Tartu University, Estonia
Fabio Martinelli	C.N.R. Pisa, Italy
Stig Mjølsnes	Norwegian University of Science and Technology
Andrei Sabelfeld	Chalmers University of Technology, Sweden
Elmar Schoch	University of Ulm, Germany

Additional Reviewers

Musard Balliu	Roberto Guancialen	Andrew Moss
Luciano Bello	Hans Hedbom	Mads Chr. Olesen
Arnar Birgisson	Sven Heiberg	Willard Rafnsson
Marco Caselli	Aivo Kalu	Daniel Schoepe
Alessio Di Mauro	Stephan Kleber	Oliver Schwarz
Stefan Dietzel	Gunnar Kreitz	Christoph Sommer
Bela Genge	Sven Laur	Rens W. van der Heijden
Madeline González Muñiz	Long-Hai Li	Jan Willemson
	Leonardo Martucci	

Towards Verified Implementation of Cryptographic Constructions Invited Talk

Gilles Barthe

IMDEA Software Institute

EasyCrypt [2] is a computer-assisted framework for reasoning about the security of cryptographic constructions in the computational model. EasyCrypt adopts the principles of provable security, and allows building reductionist proofs showing that the probability that an adversary breaks the security of the cryptographic system in "reasonable time" is "small", provided the probability that a probabilistic algorithm solves a computationally intractable problem in "reasonable time" is also "small". Over the last years, we have used EasyCrypt and its predecessor CertiCrypt to prove security of several emblematic constructions, including public-key encryption and signature schemes, modes of operations, hash designs, zero-knowledge protocols, and differentially private algorithms.

Following an established trend, EasyCrypt takes a language-based approach to provable security. Security notions and cryptographic constructions are modelled using a core probabilistic programming language, featuring sequential composition, conditionals, loops, procedure calls, deterministic assignments and probabilistic assignments drawn from discrete distributions. Thanks to their well-defined semantics, programming languages provide a natural framework to reason formally about security of cryptographic constructions. Specifically, proofs of security are executed using program logics. Because reductionist arguments reason about the execution of two programs, they cannot be captured by traditional program logics, which can only establish properties of program executions. Therefore, EasyCrypt features a Hoare logic to bound the probability of events in programs, and a relational Hoare logic that allows users to relate the probability of two events in different programs. In combination, these logics capture the most common patterns of reasoning that arise in cryptographic proofs. Using an ambient logic, one can then prove concrete security of a cryptographic construction by combining the probability claims derived from valid Hoare and relational Hoare judgments.

However, there is a significant gap between the formally verified algorithms, and their realizations in the real world. In fact, many practical attacks exploit implementation details, for instance error management or message formatting, that are typically not considered in formal proofs. Therefore, our most recent work [1] provides a framework to derive security guarantees for executable code. The front-end of the framework is an extension of EasyCrypt for reasoning about C-like programs extended with idealized probabilistic operations, such as uniform sampling or random oracles, in the style of code-based security proofs. This ex-

tension allows proving concrete security of reference implementations based on standards; it also narrows a painful gap between provable security, which considers algorithmic descriptions of the schemes, and cryptographic practice, based on implementation of standards. The back-end of the framework is based on an extension of CompCert, a verified optimizing compiler for C [3], and allows the security guarantees established at C-level to be carried over to executable code. We have applied the framework to verify the RSA-OAEP encryption scheme, as standardized in PKCS#1 v2.1.

More information about the project is available from the project web page

`http://www.easycrypt.info`

References

1. José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, and Francois Dupressoir. Certified computer-aided cryptography: efficient provably secure machine code from high-level implementations. Cryptology ePrint Archive, Report 2013/316, 2013. To appear in Proceedings of ACM Conference on Computer and Communications Security, 2013.
2. Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella-Béguelin. Computer-aided security proofs for the working cryptographer. In *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 71–90, Heidelberg, 2011. Springer.
3. X. Leroy. Formal certification of a compiler back-end, or: programming a compiler with a proof assistant. In *33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2006*, pages 42–54, New York, 2006. ACM.

Table of Contents

Cyber-Physical Systems

Detecting and Preventing Beacon Replay Attacks in Receiver-Initiated MAC Protocols for Energy Efficient WSNs 1
Alessio Di Mauro, Xenofon Fafoutis, Sebastian Mödersheim, and Nicola Dragoni

Security Games for Cyber-Physical Systems 17
Roberto Vigo, Alessandro Bruni, and Ender Yüksel

Prevent Session Hijacking by Binding the Session to the Cryptographic Network Credentials 33
Willem Burgers, Roel Verdult, and Marko van Eekelen

Security Policies

Inferring Required Permissions for Statically Composed Programs 51
Tero Hasu, Anya Helene Bagge, and Magne Haveraaen

SafeScript: JavaScript Transformation for Policy Enforcement 67
Mike Ter Louw, Phu H. Phung, Rohini Krishnamurti, and Venkat N. Venkatakrishnan

Information Flow

A Logic for Information Flow Analysis of Distributed Programs 84
Musard Balliu

Dynamics and Secure Information Flow for a Higher-Order Pi-Calculus 100
Martin Pettai and Peeter Laud

Lazy Programs Leak Secrets 116
Pablo Buiras and Alejandro Russo

Security Experiences

High-Performance Qualified Digital Signatures for X-Road 123
Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson

Identification and Evaluation of Security Activities in Agile Projects 139
Tigist Ayalew, Tigist Kidane, and Bengt Carlsson

`PeerShare`: A System Secure Distribution of Sensitive Data among Social Contacts 154
Marcin Nagy, N. Asokan, and Jörg Ott

Cyber-Physical Systems

Resilience of Process Control Systems to Cyber-Physical Attacks 166
Marina Krotofil and Alvaro A. Cárdenas

Femtocell Security in Theory and Practice 183
Fabian van den Broek and Ronny Wichers Schreur

Security Analysis of Building Automation Networks: Threat Model and Viable Mitigation Techniques 199
Alessio Antonini, Alessandro Barenghi, and Gerardo Pelosi

Web Security

Controlling Data Flow with a Policy-Based Programming Language for the Web 215
Thierry Sans, Iliano Cervesato, and Soha Hussein

A Survey on Control-Flow Integrity Means in Web Application Frameworks 231
Bastian Braun, Christian v. Pollak, and Joachim Posegga

Security Policies

Incremental Hyperproperty Model Checking via Games 247
Dimiter Milushev and Dave Clarke

Graph k-Anonymity through k-Means and as Modular Decomposition 263
Klara Stokes

Network Security

Domain-Based Storage Protection (DBSP) in Public Infrastructure Clouds 279
Nicolae Paladi, Christian Gehrmann, and Fredric Morenius

An Adaptive Mitigation Framework for Handling Suspicious Network Flows via MPLS Policies 297
Nabil Hachem, Joaquin Garcia-Alfaro, and Hervé Debar

Keyless Signatures' Infrastructure: How to Build Global Distributed Hash-Trees 313
Ahto Buldas, Andres Kroonmaa, and Risto Laanoja

Author Index 321