Information Security and Cryptography

Series Editors

David Basin
Kenny Paterson

Advisory Board

Michael Backes
Gilles Barthe
Ronald Cramer
Ivan Damgård
Andrew D. Gordon
Joshua D. Guttman
Ueli Maurer
Tatsuaki Okamoto
Bart Preneel

For further volumes:
www.springer.com/series/4752

Cas Cremers • Sjouke Mauw

Operational Semantics and Verification of Security Protocols

Springer

Cas Cremers
Department of Computer Science
ETH Zürich
Zürich, Switzerland

Sjouke Mauw
Computer Science and
Communications Research Unit
University of Luxembourg
Luxembourg, Luxembourg

ISSN 1619-7100 Information Security and Cryptography
ISBN 978-3-540-78635-1 ISBN 978-3-540-78636-8 (eBook)
DOI 10.1007/978-3-540-78636-8
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012952288

ACM Computing Classification (1998): C.2, K.6, E.3

© Springer-Verlag Berlin Heidelberg 2012
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Preface

About ten years ago, Sjouke Mauw and Erik de Vink established the computer security research group at the Eindhoven University of Technology. Given their background in formal methods, they focused on the formal modeling of security protocols and their properties. The underlying assumption was that reasoning in a relatively simple model, based on well-understood notions from the area of operational semantics, would help to understand the complexities inherent to security protocols. Soon after starting this research, Cas Cremers joined the group and took up the challenge to further develop these ideas for his PhD thesis.

Over the years, we have used the resulting model not only for theoretical research, but also for teaching the fundaments of security protocols to master students. During this time we developed the Scyther tool. The efficient tool support enabled us to quickly prove or disprove hypotheses, and enabled us to apply our approach to practical protocols.

The purpose of this book is threefold. First, it aims to present our approach in such a way that it can be used as an introduction to conducting theoretical research on security protocols. Second, it is designed to serve as a basis for teaching courses on protocol verification. Third, by providing the theoretical foundations underlying our approach, it can aid in practical protocol analysis. We hope that after reading this book, the reader has gained an understanding of the inner workings and possibilities of formal protocol analysis.

This book would have not been possible without the support of many people. First and foremost we express our great appreciation to Erik de Vink. He contributed to the conception and development of many of our technical results and we have greatly benefited from our stimulating discussions.

We are grateful to Jos Baeten and David Basin, who provided us with the opportunity to work on this research topic and the subsequent book for several years. We also express our gratitude to our editor, Ronan Nugent, who patiently provided us with support throughout the writing of this book.

The theory developed in this book and its presentation have undergone many changes over the years. We are grateful to the following people for helping to shape our ideas into their current form: David Basin, Ton van Deursen, Hugo Jonker, Bar-

bara Kordy, Simon Meier, Matthijs Melissen, Marko Horvat, Saša Radomirović, Benedikt Schmidt, and Christoph Sprenger.

Finally, our families provided moral support throughout the writing of this book. Without them, this book would not have seen the light of day.

Zürich, Cas Cremers
Luxembourg Sjouke Mauw

Contents

1 Introduction 1
- 1.1 Historical Context 1
- 1.2 Black-Box Security Protocol Analysis 3
- 1.3 Purpose and Approach 6
- 1.4 Overview 6
  - 1.4.1 The Protocol Analysis Model 6
  - 1.4.2 Applications of the Model 7

2 Preliminaries 9
- 2.1 Sets and Relations 9
- 2.2 BNF Grammars 10
- 2.3 Labelled Transition Systems 11

3 Operational Semantics 13
- 3.1 Analysis of the Problem Domain 13
- 3.2 Security Protocol Specification 17
  - 3.2.1 Role Terms 18
  - 3.2.2 Protocol Specification 21
  - 3.2.3 Event Order 23
- 3.3 Describing Protocol Execution 24
  - 3.3.1 Runs 25
  - 3.3.2 Matching 27
  - 3.3.3 Run Events 29
  - 3.3.4 Threat Model 30
- 3.4 Operational Semantics 31
- 3.5 Example Protocol Specification 33
- 3.6 Problems 34

4 Security Properties 37
- 4.1 Security Properties as Claim Events 37
- 4.2 Secrecy 39
- 4.3 Authentication 41

4.3.1 Aliveness . . . 41
4.3.2 Synchronisation . . . 45
4.3.3 Non-injective Synchronisation . . . 46
4.3.4 Injective Synchronisation . . . 48
4.3.5 Message Agreement . . . 50
4.4 Authentication Hierarchy . . . 51
4.5 Breaking and Fixing the Needham-Schroeder Protocol . . . 55
4.6 Summary . . . 62
4.7 Problems . . . 63

5 Verification . . . 67
5.1 Patterns . . . 67
5.2 Verification Algorithm . . . 74
5.2.1 Well-Typed Patterns . . . 75
5.2.2 Realisable Patterns . . . 75
5.2.3 Empty Patterns and Redundant Patterns . . . 77
5.2.4 Algorithm Overview . . . 78
5.2.5 Pattern Refinement . . . 79
5.3 Example of Search Space Traversal . . . 82
5.4 Verifying Security Properties Using Pattern Refinement . . . 89
5.5 Heuristics and Parameter Choices . . . 90
5.5.1 Heuristics . . . 90
5.5.2 Choosing a Bound on the Number of Runs . . . 92
5.5.3 Performance . . . 94
5.6 Verifying Injectivity . . . 96
5.6.1 Injective Synchronisation . . . 96
5.6.2 The LOOP Property . . . 99
5.6.3 Model Assumptions . . . 103
5.7 Further Features of the Scyther Tool . . . 103
5.8 Problems . . . 105

6 Multi-protocol Attacks . . . 107
6.1 Multi-protocol Attacks . . . 108
6.2 Experiments . . . 109
6.3 Results . . . 110
6.3.1 Strict Type Matching: No Type Flaws . . . 112
6.3.2 Simple Type Matching: Basic Type Flaws Only . . . 113
6.3.3 Untyped Matching: All Type Flaws . . . 113
6.3.4 Attack Example . . . 113
6.4 Attack Scenarios . . . 115
6.4.1 Protocol Updates . . . 116
6.4.2 Ambiguous Authentication . . . 118
6.5 Preventing Multi-protocol Attacks . . . 120
6.6 Summary . . . 122
6.7 Problems . . . 122

7 Generalising NSL for Multi-party Authentication 123
7.1 A Multi-party Authentication Protocol 124
7.2 Analysis 127
7.2.1 Initial Observations 127
7.2.2 Proof of Correctness 127
7.2.3 Secrecy of Nonces Created in Role r_0^p 132
7.2.4 Non-injective Synchronisation of Role $r0$ 133
7.2.5 Secrecy of Nonces Created in Role r_x^p for $x > 0$ 134
7.2.6 Non-injective Synchronisation of Role r_x^p for $x > 0$ 134
7.2.7 Injective Synchronisation of All Roles 135
7.2.8 Type-Flaw Attacks 135
7.2.9 Message Minimality 136
7.3 Variations on the Pattern 136
7.4 Weaker Multi-party Authentication Protocols 139
7.5 Problems 140

8 Historical Background and Further Reading 143
8.1 Historical Background 143
8.1.1 Models 143
8.1.2 Initial Tools 143
8.1.3 Logics 144
8.1.4 Tool Proliferation 145
8.1.5 Multi-protocol Attacks 146
8.1.6 Complexity Results 147
8.1.7 Divergence Between Symbolic and Computational Models 147
8.1.8 Bridging the Gap Between Symbolic Analysis and Code . . 148
8.2 Alternative Approaches 149
8.2.1 Modelling Frameworks 149
8.2.2 Security Properties 150
8.2.3 Tools 153

References 157

Index 167

List of Figures

Fig. 1.1 Scytale . . . 2
Fig. 1.2 Unfolded Scytale . . . 2
Fig. 3.1 A simple communication protocol . . . 14
Fig. 3.2 A security protocol and its execution . . . 15
Fig. 3.3 The Needham-Schroeder public-key authentication protocol . . . 17
Fig. 3.4 Overview of the specification of the *NS* protocol . . . 33
Fig. 4.1 A protocol with a correct and an incorrect claim . . . 38
Fig. 4.2 An attack on the secrecy claim of the responder role . . . 38
Fig. 4.3 Protocol $HELLO_0$ . . . 41
Fig. 4.4 Protocol $HELLO_1$ . . . 43
Fig. 4.5 Protocol $HELLO_2$ . . . 43
Fig. 4.6 Protocol $HELLO_3$ . . . 44
Fig. 4.7 An attack on the $HELLO_1$ protocol . . . 44
Fig. 4.8 Synchronisation does not hold for the $HELLO_3$ protocol . . . 45
Fig. 4.9 Protocol $HELLO_4$ . . . 47
Fig. 4.10 Message forwarding versus synchronisation . . . 48
Fig. 4.11 A protocol vulnerable to a replay attack . . . 49
Fig. 4.12 A replay attack . . . 49
Fig. 4.13 Hierarchy of authentication properties . . . 52
Fig. 4.14 A protocol that is not injective . . . 53
Fig. 4.15 A protocol that satisfies agreement but does not synchronise . . . 53
Fig. 4.16 An exploitable protocol that does not satisfy synchronisation . . . 54
Fig. 4.17 A protocol that does not satisfy non-injective agreement . . . 55
Fig. 4.18 Needham-Schroeder public-key authentication protocol . . . 56
Fig. 4.19 Attack on Needham-Schroeder . . . 57
Fig. 4.20 The Needham-Schroeder-Lowe public-key authentication protocol 58
Fig. 5.1 Example trace with adversary events . . . 69
Fig. 5.2 Pattern describing an arbitrary execution of the send and receive events of the r role of the *NS* protocol . . . 71
Fig. 5.3 Pattern describing all traces in which the secrecy claim of variable W in role r of the *NS* protocol is violated . . . 72

Fig. 5.4 The correspondence between events from the pattern from Example 5.5 (*on the left*) and the events from the trace (*on the right*) representing a regular execution of the *NS* protocol 73
Fig. 5.5 Realisable pattern of the *NS* protocol 77
Fig. 5.6 Lowe's modified version of the Yahalom protocol 84
Fig. 5.7 Search tree overview 85
Fig. 5.8 First part of the search tree 86
Fig. 5.9 Second part of search tree 87
Fig. 5.10 Third part of search tree 88
Fig. 5.11 The impact of the heuristics on the number of states traversed . . . 92
Fig. 5.12 The impact of the heuristics on the decidability 93
Fig. 5.13 The impact of the bound on runs on decidability for the protocols in the test set 94
Fig. 5.14 The impact of the bound on runs on the total verification time for 518 claims 95
Fig. 5.15 A protocol that is not vulnerable to replay attacks 96
Fig. 5.16 A unilateral agreement protocol, with LOOP, but not injective . . 104
Fig. 6.1 The Woo and Lam mutual authentication protocol 114
Fig. 6.2 Lowe's modified version of the Yahalom protocol 115
Fig. 6.3 A two-protocol attack 116
Fig. 6.4 Needham-Schroeder: Broken 117
Fig. 6.5 The Needham-Schroeder-Lowe protocol 118
Fig. 6.6 NSL attack using the broken variant 119
Fig. 6.7 Service 1 120
Fig. 6.8 Service 2 120
Fig. 6.9 Attack on combined services 1 and 2 121
Fig. 7.1 Four-party generalised NSL 124
Fig. 7.2 Generalised NSL pattern 125
Fig. 7.3 Four-party BKE protocol 137
Fig. 7.4 Four-party NSL secret-key protocol 138
Fig. 7.5 BNV Protocol 1, three-party version 139
Fig. 7.6 Attack on BNV Protocol 1, three-party version 140
Fig. 7.7 BNV Protocol 2, three-party version 141

List of Tables

Table 3.1 Components of the security protocol model 15
Table 3.2 Basic sets and some typical elements 18
Table 3.3 Basic run term sets and some typical elements 26
Table 3.4 Operational semantics rules 32
Table 5.1 *in* and *out* functions 69
Table 5.2 Verification times in seconds 95
Table 6.1 List of protocols tested in a multi-protocol environment 109
Table 6.2 Overview of protocol claims with multi-protocol attacks 111
Table 6.3 Influence of matching predicate 113