Bruce Christianson
Bruno Crispo
James A. Malcolm
Michael Roe (Eds.)

LNCS 4631

Security Protocols

13th International Workshop
Cambridge, UK, April 2005
Revised Selected Papers

Springer

Bruce Christianson Bruno Crispo
James A. Malcolm Michael Roe (Eds.)

Security Protocols

13th International Workshop
Cambridge, UK, April 20-22, 2005
Revised Selected Papers

Springer

Volume Editors

Bruce Christianson
University of Hertfordshire
Computer Science Department
Hatfield AL10 9AB, UK
E-mail: b.christianson@herts.ac.uk

Bruno Crispo
Vrije Universiteit
Department of Computer Science
1081 HV Amsterdam, The Netherlands
E-mail: crispo@cs.vu.nl

James A. Malcolm
University of Hertfordshire
Computer Science Department
Hatfield AL10 9AB, UK
E-mail: j.a.malcolm@herts.ac.uk

Michael Roe
Microsoft Research Ltd.
Cambridge CB3 0FB, UK
E-mail: mroe@microsoft.com

Library of Congress Control Number: 2007940529

CR Subject Classification (1998): E.3, F.2.1-2, C.2, K.6.5, J.1, K.4.1, D.4.6

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-77155-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-77155-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12201679 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4631

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

Editorial Board

Preface

Welcome to the Proceedings of the 13th International Security Protocols Workshop. As usual, our meeting in Cambridge was just the beginning. After that, position papers were revised (often more than once) and transcripts were circulated, discussed, and edited several times: our intention was not to produce a sterile record of who said what, but to share some promising lines of enquiry into interesting problems. Now we bring these proceedings to a wider audience so that you can join in.

Our theme this time was "The system likes you and wants to be your friend." Security is usually seen as making systems more difficult for humans to use. Might there be advantages to looking at security in the context of more general design problems? Perhaps those investigating the general properties of system design and those of us in the security community have more to say to each other than we thought.

Our thanks to Sidney Sussex College Cambridge for the use of their facilities, and to the University of Hertfordshire for lending us several of their staff.

Particular thanks to Johanna Hunt of the University of Hertfordshire for being our impresario and organizing everything, and to Lori Klimaszewska of the University of Cambridge Computing Service for transcribing the audio tapes (in which the "crash barriers" nearly prevented collisions).

The Security Protocols Workshop exists because you, the audience, participate. Once you have dived into these proceedings and have had some Eleatic thoughts, we expect to hear from you.

Bruce Christianson
Bruno Crispo
James Malcolm
Michael Roe

Previous Proceedings in This Series

The proceedings of previous International Workshops on Security Protocols have also been published by Springer as *Lecture Notes in Computer Science*, and are occasionally referred to in the text:

12th Workshop (2004), LNCS 3957, ISBN 3-540-40925-4
11th Workshop (2003), LNCS 3364, ISBN 3-540-28389-7
10th Workshop (2002), LNCS 2845, ISBN 3-540-20830-5
9th Workshop (2001), LNCS 2467, ISBN 3-540-44263-4
8th Workshop (2000), LNCS 2133, ISBN 3-540-42566-7
7th Workshop (1999), LNCS 1796, ISBN 3-540-67381-4
6th Workshop (1998), LNCS 1550, ISBN 3-540-65663-4
5th Workshop (1997), LNCS 1361, ISBN 3-540-64040-1
4th Workshop (1996), LNCS 1189, ISBN 3-540-63494-5

Lecture Notes in Computer Science

Sublibrary 4: Security and Cryptology

Vol. 4861: S. Qing, H. Imai, G. Wang (Eds.), Information and Communications Security. XIV, 508 pages. 2007.

Vol. 4859: K. Srinathan, C.P. Rangan, M. Yung (Eds.), Progress in Cryptology – INDOCRYPT 2007. XI, 426 pages. 2007.

Vol. 4856: F. Bao, S. Ling, T. Okamoto, H. Wang, C. Xing (Eds.), Cryptology and Network Security. XII, 283 pages. 2007.

Vol. 4833: K. Kurosawa (Ed.), Advances in Cryptology – ASIACRYPT 2007. XIV, 583 pages. 2007.

Vol. 4817: K.-H. Nam, G. Rhee (Eds.), Information Security and Cryptology - ICISC 2007. XIII, 367 pages. 2007.

Vol. 4812: P. McDaniel, S.K. Gupta (Eds.), Information Systems Security. XIII, 322 pages. 2007.

Vol. 4784: W. Susilo, J.K. Liu, Y. Mu (Eds.), Provable Security. X, 237 pages. 2007.

Vol. 4779: J.A. Garay, A.K. Lenstra, M. Mambo, R. Peralta (Eds.), Information Security. XIII, 437 pages. 2007.

Vol. 4776: N. Borisov, P. Golle (Eds.), Privacy Enhancing Technologies. X, 273 pages. 2007.

Vol. 4752: A. Miyaji, H. Kikuchi, K. Rannenberg (Eds.), Advances in Information and Computer Security. XIII, 460 pages. 2007.

Vol. 4734: J. Biskup, J. López (Eds.), Computer Security – ESORICS 2007. XIV, 628 pages. 2007.

Vol. 4727: P. Paillier, I. Verbauwhede (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2007. XIV, 468 pages. 2007.

Vol. 4691: T. Dimitrakos, F. Martinelli, P.Y.A. Ryan, S. Schneider (Eds.), Formal Aspects in Security and Trust. VIII, 285 pages. 2007.

Vol. 4677: A. Aldini, R. Gorrieri (Eds.), Foundations of Security Analysis and Design IV. VII, 325 pages. 2007.

Vol. 4657: C. Lambrinoudakis, G. Pernul, A M. Tjoa (Eds.), Trust, Privacy and Security in Digital Business. XIII, 291 pages. 2007.

Vol. 4637: C. Kruegel, R. Lippmann, A. Clark (Eds.), Recent Advances in Intrusion Detection. XII, 337 pages. 2007.

Vol. 4631: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. IX, 347 pages. 2007.

Vol. 4622: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007. XIV, 631 pages. 2007.

Vol. 4593: A. Biryukov (Ed.), Fast Software Encryption. XI, 467 pages. 2007.

Vol. 4586: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), Information Security and Privacy. XIV, 476 pages. 2007.

Vol. 4582: J. López, P. Samarati, J.L. Ferrer (Eds.), Public Key Infrastructure. XI, 375 pages. 2007.

Vol. 4579: B.M. Hämmerli, R. Sommer (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment. X, 251 pages. 2007.

Vol. 4575: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing-Based Cryptography – Pairing 2007. XI, 408 pages. 2007.

Vol. 4521: J. Katz, M. Yung (Eds.), Applied Cryptography and Network Security. XIII, 498 pages. 2007.

Vol. 4515: M. Naor (Ed.), Advances in Cryptology - EUROCRYPT 2007. XIII, 591 pages. 2007.

Vol. 4499: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security II. IX, 117 pages. 2007.

Vol. 4464: E. Dawson, D.S. Wong (Eds.), Information Security Practice and Experience. XIII, 361 pages. 2007.

Vol. 4462: D. Sauveron, K. Markantonakis, A. Bilas, J.-J. Quisquater (Eds.), Information Security Theory and Practices. XII, 255 pages. 2007.

Vol. 4450: T. Okamoto, X. Wang (Eds.), Public Key Cryptography – PKC 2007. XIII, 491 pages. 2007.

Vol. 4437: J.L. Camenisch, C.S. Collberg, N.F. Johnson, P. Sallee (Eds.), Information Hiding. VIII, 389 pages. 2007.

Vol. 4392: S.P. Vadhan (Ed.), Theory of Cryptography. XI, 595 pages. 2007.

Vol. 4377: M. Abe (Ed.), Topics in Cryptology – CT-RSA 2007. XI, 403 pages. 2006.

Vol. 4356: E. Biham, A.M. Youssef (Eds.), Selected Areas in Cryptography. XI, 395 pages. 2007.

Vol. 4341: P.Q. Nguyên (Ed.), Progress in Cryptology - VIETCRYPT 2006. XI, 385 pages. 2006.

Vol. 4332: A. Bagchi, V. Atluri (Eds.), Information Systems Security. XV, 382 pages. 2006.

Vol. 4329: R. Barua, T. Lange (Eds.), Progress in Cryptology - INDOCRYPT 2006. X, 454 pages. 2006.

Vol. 4318: H. Lipmaa, M. Yung, D. Lin (Eds.), Information Security and Cryptology. XI, 305 pages. 2006.

Vol. 4307: P. Ning, S. Qing, N. Li (Eds.), Information and Communications Security. XIV, 558 pages. 2006.

Vol. 4301: D. Pointcheval, Y. Mu, K. Chen (Eds.), Cryptology and Network Security. XIII, 381 pages. 2006.

Vol. 4300: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security I. IX, 139 pages. 2006.

Vol. 4298: J.K. Lee, O. Yi, M. Yung (Eds.), Information Security Applications. XIV, 406 pages. 2007.

Vol. 4296: M.S. Rhee, B. Lee (Eds.), Information Security and Cryptology – ICISC 2006. XIII, 358 pages. 2006.

Vol. 4284: X. Lai, K. Chen (Eds.), Advances in Cryptology – ASIACRYPT 2006. XIV, 468 pages. 2006.

Vol. 4283: Y.Q. Shi, B. Jeon (Eds.), Digital Watermarking. XII, 474 pages. 2006.

Vol. 4266: H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, S.-i. Kawamura (Eds.), Advances in Information and Computer Security. XIII, 438 pages. 2006.

Vol. 4258: G. Danezis, P. Golle (Eds.), Privacy Enhancing Technologies. VIII, 431 pages. 2006.

Vol. 4249: L. Goubin, M. Matsui (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2006. XII, 462 pages. 2006.

Vol. 4237: H. Leitold, E.P. Markatos (Eds.), Communications and Multimedia Security. XII, 253 pages. 2006.

Vol. 4236: L. Breveglieri, I. Koren, D. Naccache, J.-P. Seifert (Eds.), Fault Diagnosis and Tolerance in Cryptography. XIII, 253 pages. 2006.

Vol. 4219: D. Zamboni, C. Krügel (Eds.), Recent Advances in Intrusion Detection. XII, 331 pages. 2006.

Vol. 4189: D. Gollmann, J. Meier, A. Sabelfeld (Eds.), Computer Security – ESORICS 2006. XI, 548 pages. 2006.

Vol. 4176: S.K. Katsikas, J. López, M. Backes, S. Gritzalis, B. Preneel (Eds.), Information Security. XIV, 548 pages. 2006.

Vol. 4117: C. Dwork (Ed.), Advances in Cryptology - CRYPTO 2006. XIII, 621 pages. 2006.

Vol. 4116: R. De Prisco, M. Yung (Eds.), Security and Cryptography for Networks. XI, 366 pages. 2006.

Vol. 4107: G. Di Crescenzo, A. Rubin (Eds.), Financial Cryptography and Data Security. XI, 327 pages. 2006.

Vol. 4083: S. Fischer-Hübner, S. Furnell, C. Lambrinoudakis (Eds.), Trust and Privacy in Digital Business. XIII, 243 pages. 2006.

Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), Information Security and Privacy. XII, 446 pages. 2006.

Vol. 4047: M.J.B. Robshaw (Ed.), Fast Software Encryption. XI, 434 pages. 2006.

Vol. 4043: A.S. Atzeni, A. Lioy (Eds.), Public Key Infrastructure. XI, 261 pages. 2006.

Vol. 4004: S. Vaudenay (Ed.), Advances in Cryptology - EUROCRYPT 2006. XIV, 613 pages. 2006.

Vol. 3995: G. Müller (Ed.), Emerging Trends in Information and Communication Security. XX, 524 pages. 2006.

Vol. 3989: J. Zhou, M. Yung, F. Bao (Eds.), Applied Cryptography and Network Security. XIV, 488 pages. 2006.

Vol. 3969: Ø. Ytrehus (Ed.), Coding and Cryptography. XI, 443 pages. 2006.

Vol. 3958: M. Yung, Y. Dodis, A. Kiayias, T.G. Malkin (Eds.), Public Key Cryptography - PKC 2006. XIV, 543 pages. 2006.

Vol. 3957: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. IX, 325 pages. 2006.

Vol. 3956: G. Barthe, B. Grégoire, M. Huisman, J.-L. Lanet (Eds.), Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. IX, 175 pages. 2006.

Vol. 3935: D. H. Won, S. Kim (Eds.), Information Security and Cryptology - ICISC 2005. XIV, 458 pages. 2006.

Vol. 3934: J.A. Clark, R.F. Paige, F.A.C. Polack, P.J. Brooke (Eds.), Security in Pervasive Computing. X, 243 pages. 2006.

Vol. 3928: J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.), Smart Card Research and Advanced Applications. XI, 359 pages. 2006.

Vol. 3919: R. Safavi-Naini, M. Yung (Eds.), Digital Rights Management. XI, 357 pages. 2006.

Vol. 3903: K. Chen, R. Deng, X. Lai, J. Zhou (Eds.), Information Security Practice and Experience. XIV, 392 pages. 2006.

Vol. 3897: B. Preneel, S. Tavares (Eds.), Selected Areas in Cryptography. XI, 371 pages. 2006.

Vol. 3876: S. Halevi, T. Rabin (Eds.), Theory of Cryptography. XI, 617 pages. 2006.

Vol. 3866: T. Dimitrakos, F. Martinelli, P.Y.A. Ryan, S. Schneider (Eds.), Formal Aspects in Security and Trust. X, 259 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3858: A. Valdes, D. Zamboni (Eds.), Recent Advances in Intrusion Detection. X, 351 pages. 2006.

Vol. 3856: G. Danezis, D. Martin (Eds.), Privacy Enhancing Technologies. VIII, 273 pages. 2006.

Vol. 3786: J.-S. Song, T. Kwon, M. Yung (Eds.), Information Security Applications. XI, 378 pages. 2006.

Vol. 3108: H. Wang, J. Pieprzyk, V. Varadharajan (Eds.), Information Security and Privacy. XII, 494 pages. 2004.

Vol. 2951: M. Naor (Ed.), Theory of Cryptography. XI, 523 pages. 2004.

Vol. 2742: R.N. Wright (Ed.), Financial Cryptography. VIII, 321 pages. 2003.

Table of Contents

The System Likes You (Transcript of Discussion) 1
Bruce Christianson

Experiences with Host-to-Host IPsec 3
Tuomas Aura, Michael Roe, and Anish Mohammed*
Discussion 23

Repairing the Bluetooth Pairing Protocol 31
Ford-Long Wong, Frank Stajano, and Jolyon Clulow*
Discussion 46

Keep on Blockin' in the Free World: Personal Access Control for Low-Cost RFID Tags 51
Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum*
Discussion 60

PIN (and Chip) or Signature: Beating the Cheating? 69
*Dan Cvrcek, Jan Krhovjak, and Vashek Matyas**
Discussion 76

Insecure Real-World Authentication Protocols (or Why Phishing Is So Profitable) 82
Richard Clayton
Discussion 89

Authorisation Subterfuge by Delegation in Decentralised Networks 97
Simon Foley and Hongbin Zhou*
Discussion 103

Multi-channel Protocols 112
Ford-Long Wong and Frank Stajano*
Discussion 128

Combining Crypto with Biometrics: A New Human-Security Interface (Transcript of Discussion) 133
Feng Hao

User-Friendly Grid Security Architecture and Protocols 139
Liqun Chen, Hoon Wei Lim, and Wenbo Mao*
Discussion 157

Countering Automated Exploits with System Security CAPTCHAS 162
Dinan Gunawardena, Jacob Scott, Alf Zugenmaier, and Austin Donnelly*
Discussion 170

The System Likes You? (Transcript of Discussion) 180
Mark Lomas

Enhancing Privacy with Shared Pseudo Random Sequences 187
Jari Arkko, Pekka Nikander, and Mats Näslund*
Discussion 197

Non-repudiation and the Metaphysics of Presence (Extended Abstract) 204
Michael Roe
Discussion 207

Understanding Why Some Network Protocols Are User-Unfriendly 215
Yvo Desmedt
Discussion 220

Community-Centric Vanilla-Rollback Access, or: How I Stopped Worrying and Learned to Love My Computer 228
Mike Burmester, Breno de Medeiros, and Alec Yasinsac*
Discussion 238

Listen Too Closely and You May Be Confused 245
*Eric Cronin, Micah Sherr, and Matt Blaze**
Discussion 250

The Dining Freemasons (Security Protocols for Secret Societies) 258
Mike Bond and George Danezis*
Discussion 266

On the Evolution of Adversary Models in Security Protocols (or Know Your Friend and Foe Alike) (Transcript of Discussion) 276
Virgil Gligor

Safer Scripting Through Precompilation 284
Ben Laurie
Discussion 289

Implementing a Multi-hat PDA 295
Matthew Johnson and Frank Stajano*
Discussion 308

Anonymous Context Based Role Activation Mechanism 315
Partha Das Chowdhury, Bruce Christianson, and James Malcolm*
Discussion 322

Topology of Covert Conflict (Transcript of Discussion) 329
Shishir Nagaraja

The Initial Costs and Maintenance Costs of Protocols 333
Ross Anderson
Discussion 336

Alice and Bob 344
John Gordon

Author Index 347

The System Likes You
(Transcript of Discussion)

Bruce Christianson

Every year we have a theme and it's always difficult to ignore the theme if you don't know what it is, so there's a tradition that somebody spends five minutes at the beginning telling you what the theme is so that you can ignore it.

The theme this year is "the system likes you and wants to be your friend". The thinking behind this is that there might be advantages to looking at security in the context of more general design problems, and that those investigating the general properties of system design and those of us in the security community might have more to say to each other than we currently seem to.

We are all used to the idea that we design systems to have certain properties that we want them to have. In the security community we want to design systems also to not have certain properties (or to have certain *anti-properties*) which correspond to behaviour which we don't want the system to exhibit. These are complimentary activities in some ways, and run directly counter to each other in others.

We in the security community have tended to make a dichotomy between things that are the product of malice and things that are the product of bad luck. Roger Needham and Ross Anderson wrote a paper some time ago called "Programming Satan's Computer" [1] in which they almost argue that Satan is an adversary of a qualitatively different character to Murphy. A consequence of this view is that the techniques that we use to deal with a general kind of fault-tolerant threat are in some sense orthogonal to the techniques that we have to use to deal with a malicious opponent. So one question is, is this dichotomy still appropriate, or might we do better to treat Murphy and Satan as occupying extreme points of a spectrum, and think more about what is in the middle.[2]

In design exercises we are accustomed to designing a system by conceptually moving the system boundary around. If you're not sure how to design some part of the system then it's a good idea to try moving the system boundary across it and seeing what happens to your thinking about the rest of the system. Now that might perhaps tell us that trying to make a system secure by securing the system perimeter is not a good approach. Sometimes it may be more useful to consider the effects of moving the attacker across the system perimeter, and the same for the user or client, and for various bits of the infrastructure. I think the potential for interaction goes the other way as well. Roger Needham was fond of saying that if you had a completely secure system then at least turning cryptography on and off was a good way of checking whether you'd filled the configuration file in correctly.

[1] Computer Science Today, LNCS 1000, pp426–441.

[2] For example, see Geraint Price, Broadening the Scope of Fault Tolerance within Secure Services, LNCS 2133, pp155–169.

B. Christianson et al. (Eds.): Security Protocols 2005, LNCS 4631, pp. 1–2, 2007.

We've already had a lot of experience in the security community with the idea that different participants in the system have different security policies and hence different threat models. A consequence is that when we're considering our interactions with someone else, we have to consider states of the system that are possible according to their threat model, but which according to our threat model are not merely contingently false but impossible *per se*[3]. Often one of the other participants in the system will insert a counter-measure to counteract a threat which they believe is real, and which we believe is imaginary, but actually we believe that the threat posed to us by their counter-measure is very real indeed. We spend a lot of time in practice trying to deal with the consequences of these counter-factual counter-measures, and there's a possibility that the design community might benefit from looking at systems in this kind of way. We've already had extreme programming, perhaps paranoid design is the next logical extension.

The ground rules are the usual ones. This is a workshop not a conference, and it's perfectly OK to talk about things that you haven't done yet. But of course it's also perfectly OK for people to interject from the floor about things they haven't done yet either. Try and look on this as leading a discussion rather than giving a presentation that you prepared before you came. Equally if you're interjecting try and make sure that you do let the person who's giving the talk say something important at some point, even if it's just at tea time. The computer is working perfectly so let's begin.

[3] LNCS 1796, pp63–64.

Experiences with Host-to-Host IPsec

Tuomas Aura, Michael Roe, and Anish Mohammed

Microsoft Research

Abstract. This paper recounts some lessons that we learned from the deployment of host-to-host IPsec in a large corporate network. Several security issues arise from mismatches between the different identifier spaces used by applications, by the IPsec security policy database, and by the security infrastructure (X.509 certificates or Kerberos). Mobile hosts encounter additional problems because private IP addresses are not globally unique, and because they rely on an untrusted DNS server at the visited network. We also discuss a feature interaction in an enhanced IPsec firewall mechanism. The potential solutions are to relax the transparency of IPsec protection, to put applications directly in charge of their security and, in the long term, to redesign the security protocols not to use IP addresses as host identifiers.

1 Introduction

IPsec is a network-layer security protocol for the Internet that is intended to provide authentication and encryption of IP packets in a way that is transparent to applications. IPsec can be used between two hosts, between two security gateways, or between a host and a security gateway. IPsec was primarily specified with the security gateways and virtual private networks (VPN) in mind, but the expectation was that it could also be used end-to-end between two hosts.

This paper explains some of the difficulties that arise when IPsec is used in a host-to-host setting. The paper is based on security analysis and experiments that were done during the deployment of host-to-host IPsec on a large production network (tens of thousands of Windows hosts). We believe that the problems discovered are not unique to one network or one vendor's implementation, and that they explain why there are few examples of successful host-to-host IPsec deployments in large or medium-size networks.

The problems identified in this paper arise mainly from mismatches between the identifier spaces used in applications, in the IP layer, and in the security infrastructure. For example, IPsec security policies are typically defined in terms of IP addresses but the addresses mean little to the application and they do not appear in authentication credentials. Another reason for the problems is the fundamental design principle in IPsec that it should be a transparent layer that has no interaction with applications, apart from the configuration of a static security policy at the time of setting up the applications. We show that, in order for IPsec to meet the real application security requirements, the transparency needs to be relaxed and applications need to become security aware.

B. Christianson et al. (Eds.): Security Protocols 2005, LNCS 4631, pp. 3–22, 2007.

Most literature on security protocols concentrates on cryptographic algorithms and key-exchange protocols. We now know how to engineer a security protocol for authentication and encryption between abstract entities like Initiator and Respondent, or Alice and Bob. The latest IPsec specifications benefit from this work and represent the state of art in the field. The focus of this paper is on architectural issues, such as who defines the security policies and who has the authority over the various identifier spaces. We assume that the algorithms and protocols themselves are sound.

Arguments can be made that the vulnerabilities described in this paper are caused by flaws in the IPsec architecture. Equally well, it can be argued that we are using IPsec in the wrong way or that it has been implemented incorrectly. Either way, end-to-end encryption and authentication between hosts belonging to the same organization is clearly a reasonable security mechanism to ask for, and IPsec is a reasonable candidate to consider for the task. If IPsec in its currently implemented form fails, as we demonstrate it does, it then makes sense to ask what changes are needed to make the architecture meet our requirements.

The rest of the paper is structured as follows. We start with an overview of related work in section 2. Sections 3-4 provide an introduction to the IPsec architecture and to a well-known class of DNS-spoofing attacks. Section 5 shows how similar attacks are possible against host-to-host IPsec even if the name service is assumed to be secure. In section 6, we present a class of attacks that affects mobile hosts. Section 7 discusses an attack that was made possible by a non-standard extension to IPsec, and section 8 concludes the paper.

2 Related Work

IPsec is defined by the Internet Engineering Task Force (IETF). The earlier IPsec architecture specification [11] was based on early implementation experiences. The latest revised version [12] has a well-defined security model. There are also two versions of the Internet key exchange protocol, IKEv1 [10] and IKEv2 [7]. Where it matters, we use the latest specifications. All observations and experiments, however, were conducted with implementations that follow the older specifications.

The research community has paid a lot of attention to the cryptography and the key-exchange protocol in IPsec [3,8,13,19]. There is also some work on security-policy specification in large systems [9]. The architecture itself has received surprisingly little attention outside the IETF. The closest precedent to our work is by Ferguson and Schneier [8] who, in addition to evaluating the cryptography, make some radical recommendations for changes to the overall architecture, such as elimination of the transport mode. While the focus of this paper is on transport mode, all our observations apply equally well to tunnel mode when it is used host-to-host. Ferguson and Schneier also suggest that using IPsec for anything other than VPN will lead to problems but they do not elaborate on the topic. Meadows [16] pointed out that there might be problems with IKE if the authenticated identifiers are not based on IP addresses. A

yet-unpublished article by Trostle and Grossman [23] also discusses identifier-space issues in IPsec that are similar to the ones we raise in section 5.

Recently, much attention has been paid to "weak" security mechanisms, such as cookie exchanges, which reduce the number of potential attackers or make the attacks more expensive [2,17,22]. While these solutions are suitable for many applications and, in particular, for denial-of-service (DoS) prevention, they do not provide the kind of strong encryption and authentication that are the goal of IPsec. Thus, we have to assume a Dolev-Yao-type attacker [5] and cannot argue that a vulnerability in IPsec does not matter because it is unlikely that the attacker will be in the right place at the right time to exploit it. We do, however, compare the consequences of having a secure (e.g., DNSSec [6]) and an insecure name service. The conclusions are valid regardless of how the name-service security is implemented.

One high-level explanation for some of the problems discussed in this paper is that IP addresses are both host identifiers and location addresses [4]. There have been several attempts to separate these two functions, including the host identity protocol (HIP) [18], and Cryptographically Generated Addresses (CGA) [1]. In HIP, routable IP addresses are used only for routing and hosts are identified by the hash of a public key. Cryptographically generated addresses (CGA) [1], on the other hand, are routable IPv6 addresses that have the hash of a public key embedded in the address bits. This makes them work better as host identifiers. Mismatches between the identifiers used to specify the security policy and the identifiers provided by the authentication protocol have been studied in other protocol layers, for example middleware [14]. While we believe that many of these approaches have merit, we have chosen to work with standard DNS names and IP addresses in this paper.

3 How IPsec Works

In this section, we give a simplified overview of the IPsec architecture with emphasis on the features that are needed in the rest of the paper. The architecture comprises protocols, security associations, and security policies.

IPsec, like most network security protocols, has a session protocol for the protection of data, and an authenticated key exchange protocol for establishing a shared session key. The session protocol is called the *Encapsulating Security Payload* (ESP). It takes care of the encryption and/or authentication of individual IP packets. There is another session protocol, the *Authentication Header* (AH), but its use is no longer recommended. The session keys are negotiated with the Internet Key Exchange (IKE), of which there are two versions (IKEv1 and IKEv2). The differences between the versions are largely unimportant to this paper.

The shared session state between two IPsec nodes is called a *security association* (SA). An SA determines the session protocol mode, the cryptographic algorithms, and the session keys used between the nodes. The SAs come in pairs,

one for each direction. Security associations are typically created by IKE, but they can also be configured manually by the system administrator.

In addition to the protocols and associations, an important part of the IPsec architecture is the security policy. Each host has a *security policy database* (SPD) that determines an *action* for each packet: whether the packet should be protected, discarded, or allowed to bypass IPsec processing. The SPD maps the protected packets to the right SAs, and triggers IKE to create an SA pair if no suitable one exists. The policy applies to both outbound and inbound packets. For outbound packets, an SA is used to add the encryption and message authentication code as required by the policy. For inbound packets, the policy determines what kind of protection the packet must have. Inbound packets that do not have the right protection, i.e., ones that were not received via the right SA, are discarded by IPsec.

The SPD is an ordered list of rules, each one of which consists of *selectors* and an action. The packet headers are compared against the selectors and the first rule with matching selectors determines the action to be taken on the packet. The exact packet-matching algorithm has changed over versions of the IPsec specification and varies from implementation to implementation; we stick to what is common between many implementations. The selectors are typically ranges of packet-header values, e.g., source and destination IP addresses and port numbers. For example, the SPD of figure 1 mandates ESP protection for communication with peers in the subnet 1.2.*.* and a BYPASS policy for other peers. In theory, the selectors are not limited to IP and transport-layer header fields but can take into account other context, such as the hostname, process and user at the source or destination. In practice, such selectors are rarely implemented because they can cause layer violations and are, therefore, hard to implement. (We will return to names as selectors in section 6.4.)

In this paper, we are interested in the difference between two types of IPsec applications. The first application is a VPN, in which encrypted and authenticated tunnels connect geographically separate parts of a private network. Originally, IPsec tunnels over the Internet were used to replace leased telephone lines and the goal was to provide security equivalent to a leased line. The tunnel in that case is an IPsec SA between two security gateways. Increasingly, VPN technology is used to connect individual remote hosts to private networks. In that case, an IPsec SA between a remote host and a security gateway replaces a dialup connection. In both situations, the SA is set up in tunnel mode. When establishing a tunnel-mode SA, authentication in IKE is typically based on a pre-shared long-term key or X.509 certificates [24] issued by an organizational certification authority.

The second IPsec application is host-to-host communication. In that case, the IPsec SAs are established between end hosts and encryption and authentication are performed by the end hosts themselves. This kind of SA is usually set up in transport mode, although tunnel mode can also be used. The modes have subtle differences in packet headers and in policy lookup, but we need not consider them here. The host-to-host communication differs from VPNs in that one uniform