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

LNCS 2133

Security Protocols

8th International Workshop
Cambridge, UK, April 2000
Revised Papers

Springer

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

Security Protocols

8th International Workshop
Cambridge, UK, April 3-5, 2000
Revised Papers

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Bruce Christianson
James A. Malcolm
University of Hertfordshire, Computer Science Department
Hatfield AL10 9AB, UK
E-mail: {b.christianson/J.A.Malcolm}@herts.ac.uk

Bruno Crispo
Cryptomathic
Corso Svizzera 185, 10149 Torino, Italy
E-mail: bc201@cl.cam.ac.uk

Michael Roe
Microsoft Research Ltd.
St. George House, 1 Guildhall Street, Cambridge CB2 3NH, UK
E-mail: mroe@microsoft.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Security protocols : 8th international workshop, Cambridge, UK, April 3 - 5,
2000 ; revised papers / Bruce Christianson ... (ed.). - Berlin ; Heidelberg ;
New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo :
Springer, 2001
(Lecture notes in computer science ; Vol. 2133)
ISBN 3-540-42566-7

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

ISSN 0302-9743
ISBN 3-540-42566-7 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN: 10840096 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2133
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Preface

The Cambridge International Workshop on Security Protocols has now run for eight years. Each year we set a theme, focusing upon a specific aspect of security protocols, and invite position papers. Anybody is welcome to send us a position paper (yes, you are invited) and we don't insist they relate to the current theme in an obvious way. In our experience, the emergence of the theme as a unifying thread takes place during the discussions at the workshop itself. The only ground rule is that position papers should formulate an approach to some unresolved issues, rather than being a description of a finished piece of work.

When the participants meet, we try to focus the discussions upon the conceptual issues which emerge. Security protocols link naturally to many other areas of Computer Science, and deep water can be reached very quickly. Afterwards, we invite participants to re-draft their position papers in a way which exposes the emergent issues but leaves open the way to their further development. We also prepare written transcripts of the recorded discussions. These are edited (in some cases very heavily) to illustrate the way in which the different arguments and perspectives have interacted.

We publish these proceedings as an invitation to the research community. Although many interesting results first see the light of day in a volume of our proceedings, laying claim to these is not our primary purpose of publication. Rather, we bring our discussions and insights to a wider audience in order to suggest new lines of investigation which the community may fruitfully pursue.

This year's theme is "Broadening the Protocol Boundary". The boundary of a security protocol has traditionally been drawn very narrowly. Many security protocol "failures" involve factors that were not considered part of the protocol, such as the user interface. In addition, security protocols operate in a naturally fragile environment, and not all threats involve malice on the part of an attacker. Where did Alice get the information she sent, and what is Bob going to do with it? Who and what are the protocol end-points, and which domains are they in?

We invite you to consider these issues with us as you read these proceedings. See you next year, perhaps?

July 2001

Bruce Christianson
Bruno Crispo
James Malcolm
Michael Roe

Acknowledgements

Thanks to Professor Stewart Lee and the University of Cambridge Centre for Communications Systems Research who acted as hosts for the workshop, and to Professor Roger Needham FRS and Microsoft Research Limited (Cambridge) who provided us with the use of their meeting room and coffee machine. Plaudits of gratitude also to Dorian Addison of CCSR and to Angela Leeke and Margaret Nicell of MSRL for impeccable organization and administration, to Lori Klimaszewska of the University of Cambridge Computing Service for transcribing the audio tapes (including the static which changed a word attachment into a work of passion) and to Dr Mary Buchannan for her Procrustean editorial assistance.

Previous Proceedings in this Series

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

For information about Vols. 1–2086
please contact your bookseller or Springer-Verlag

Vol. 1982: S. Näher, D. Wagner (Eds.), Algorithm Engineering. Proceedings, 2000. VIII, 243 pages. 2001.

Vol. 1996: P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Advances in Learning Classifier Systems. Proceedings, 2000. VIII, 273 pages. 2001. (Subseries LNAI).

Vol. 2087: G. Kern-Isberner, Conditionals in Nonmonotonic Reasoning and Belief Revision. X, 190 pages. 2001. (Subseries LNAI).

Vol. 2088: S. Yu, A. Păun (Eds.), Implementation and Application of Automata. Proceedings, 2000. XI, 343 pages. 2001.

Vol. 2089: A. Amir, G.M. Landau (Eds.), Combinatorial Pattern Matching. Proceedings, 2001. VIII, 273 pages. 2001.

Vol. 2090: E. Brinksma, H. Hermanns, J.-P. Katoen (Eds.), Lectures on Formal Methods and Performance Analysis. Proceedings, 2000. VII, 431 pages. 2001.

Vol. 2091: J. Bigun, F. Smeraldi (Eds.), Audio- and Video-Based Biometric Person Authentication. Proceedings, 2001. XIII, 374 pages. 2001.

Vol. 2092: L. Wolf, D. Hutchison, R. Steinmetz (Eds.), Quality of Service – IWQoS 2001. Proceedings, 2001. XII, 435 pages. 2001.

Vol. 2093: P. Lorenz (Ed.), Networking – ICN 2001. Proceedings, 2001. Part I. XXV, 843 pages. 2001.

Vol. 2094: P. Lorenz (Ed.), Networking – ICN 2001. Proceedings, 2001. Part II. XXV, 899 pages. 2001.

Vol. 2095: B. Schiele, G. Sagerer (Eds.), Computer Vision Systems. Proceedings, 2001. X, 313 pages. 2001.

Vol. 2096: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. Proceedings, 2001. XII, 456 pages. 2001.

Vol. 2097: B. Read (Ed.), Advances in Databases. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2098: J. Akiyama, M. Kano, M. Urabe (Eds.), Discrete and Computational Geometry. Proceedings, 2000. XI, 381 pages. 2001.

Vol. 2099: P. de Groote, G. Morrill, C. Retoré (Eds.), Logical Aspects of Computational Linguistics. Proceedings, 2001. VIII, 311 pages. 2001. (Subseries LNAI).

Vol. 2100: R. Küsters, Non-Standard Inferences in Description Logocs. X, 250 pages. 2001. (Subseries LNAI).

Vol. 2101: S. Quaglini, P. Barahona, S. Andreassen (Eds.), Artificial Intelligence in Medicine. Proceedings, 2001. XIV, 469 pages. 2001. (Subseries LNAI).

Vol. 2102: G. Berry, H. Comon, A. Finkel (Eds.), Computer-Aided Verification. Proceedings, 2001. XIII, 520 pages. 2001.

Vol. 2103: M. Hannebauer, J. Wendler, E. Pagello (Eds.), Balancing Reactivity and Social Deliberation in Multi-Agent Systems. VIII, 237 pages. 2001. (Subseries LNAI).

Vol. 2104: R. Eigenmann, M.J. Voss (Eds.), OpenMP Shared Memory Parallel Programming. Proceedings, 2001. X, 185 pages. 2001.

Vol. 2105: W. Kim, T.-W. Ling, Y-J. Lee, S.-S. Park (Eds.), The Human Society and the Internet. Proceedings, 2001. XVI, 470 pages. 2001.

Vol. 2106: M. Kerckhove (Ed.), Scale-Space and Morphology in Computer Vision. Proceedings, 2001. XI, 435 pages. 2001.

Vol. 2107: F.T. Chong, C. Kozyrakis, M. Oskin (Eds.), Intelligent Memory Systems. Proceedings, 2000. VIII, 193 pages. 2001.

Vol. 2108: J. Wang (Ed.), Computing and Combinatorics. Proceedings, 2001. XIII, 602 pages. 2001.

Vol. 2109: M. Bauer, P.J. Gymtrasiewicz, J. Vassileva (Eds.), User Modelind 2001. Proceedings, 2001. XIII, 318 pages. 2001. (Subseries LNAI).

Vol. 2110: B. Hertzberger, A. Hoekstra, R. Williams (Eds.), High-Performance Computing and Networking. Proceedings, 2001. XVII, 733 pages. 2001.

Vol. 2111: D. Helmbold, B. Williamson (Eds.), Computational Learning Theory. Proceedings, 2001. IX, 631 pages. 2001. (Subseries LNAI).

Vol. 2116: V. Akman, P. Bouquet, R. Thomason, R.A. Young (Eds.), Modeling and Using Context. Proceedings, 2001. XII, 472 pages. 2001. (Subseries LNAI).

Vol. 2117: M. Beynon, C.L. Nehaniv, K. Dautenhahn (Eds.), Cognitive Technology: Instruments of Mind. Proceedings, 2001. XV, 522 pages. 2001. (Subseries LNAI).

Vol. 2118: X.S. Wang, G. Yu, H. Lu (Eds.), Advances in Web-Age Information Management. Proceedings, 2001. XV, 418 pages. 2001.

Vol. 2119: V. Varadharajan, Y. Mu (Eds.), Information Security and Privacy. Proceedings, 2001. XI, 522 pages. 2001.

Vol. 2120: H.S. Delugach, G. Stumme (Eds.), Conceptual Structures: Broadening the Base. Proceedings, 2001. X, 377 pages. 2001. (Subseries LNAI).

Vol. 2121: C.S. Jensen, M. Schneider, B. Seeger, V.J. Tsotras (Eds.), Advances in Spatial and Temporal Databases. Proceedings, 2001. XI, 543 pages. 2001.

Vol. 2123: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition. Proceedings, 2001. XI, 363 pages. 2001. (Subseries LNAI).

Vol. 2124: W. Skarbek (Ed.), Computer Analysis of Images and Patterns. Proceedings, 2001. XV, 743 pages. 2001.

Vol. 2125: F. Dehne, J.-R. Sack, R. Tamassia (Eds.), Algorithms and Data Structures. Proceedings, 2001. XII, 484 pages. 2001.

Vol. 2126: P. Cousot (Ed.), Static Analysis. Proceedings, 2001. XI, 439 pages. 2001.

Vol. 2127: V. Malyshkin (Ed.), Parallel Computing Technologies. Proceedings, 2001. XII, 516 pages. 2001.

Vol. 2129: M. Goemans, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization, and Combinatorial Optimization. Proceedings, 2001. IX, 297 pages. 2001.

Vol. 2130: G. Dorffner, H. Bischof, K. Hornik (Eds.), Artificial Neural Networks – ICANN 2001. Proceedings, 2001. XXII, 1259 pages. 2001.

Vol. 2132: S.-T. Yuan, M. Yokoo (Eds.), Intelligent Agents. Specification. Modeling, and Application. Proceedings, 2001. X, 237 pages. 2001. (Subseries LNAI).

Vol. 2133: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. Proceedings, 2001. VIII, 257 pages. 2001.

Vol. 2136: J. Sgall, A. Pultr, P. Kolman (Eds.), Mathematical Foundations of Computer Science 2001. Proceedings, 2001. XII, 716 pages. 2001.

Vol. 2138: R. Freivalds (Ed.), Fundamentals of Computation Theory. Proceedings, 2001. XIII, 542 pages. 2001.

Vol. 2139: J. Kilian (Ed.), Advances in Cryptology – CRYPTO 2001. Proceedings, 2001. XI, 599 pages. 2001.

Vol. 2141: G.S. Brodal, D. Frigioni, A. Marchetti-Spaccamela (Eds.), Algorithm Engineering. Proceedings, 2001. X, 199 pages. 2001.

Vol. 2142: L. Fribourg (Ed.), Computer Science Logic. Proceedings, 2001. XII, 615 pages. 2001.

Vol. 2143: S. Benferhat, P. Besnard (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Proceedings, 2001. XIV, 818 pages. 2001. (Subseries LNAI).

Vol. 2144: T. Margaria, T. Melham (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 2001. XII, 482 pages. 2001.

Vol. 2146: J.H. Silverman (Eds.), Cryptography and Lattices. Proceedings, 2001. VII, 219 pages. 2001.

Vol. 2147: G. Brebner, R. Woods (Eds.), Field-Programmable Logic and Applications. Proceedings, 2001. XV, 665 pages. 2001.

Vol. 2149: O. Gascuel, B.M.E. Moret (Eds.), Algorithms in Bioinformatics. Proceedings, 2001. X, 307 pages. 2001.

Vol. 2150: R. Sakellariou, J. Keane, J. Gurd, L. Freeman (Eds.), Euro-Par 2001 Parallel Processing. Proceedings, 2001. XXX, 943 pages. 2001.

Vol. 2151: A. Caplinskas, J. Eder (Eds.), Advances in Databases and Information Systems. Proceedings, 2001. XIII, 381 pages. 2001.

Vol. 2152: R.J. Boulton, P.B. Jackson (Eds.), Theorem Proving in Higher Order Logics. Proceedings, 2001. X, 395 pages. 2001.

Vol. 2153: A.L. Buchsbaum, J. Snoeyink (Eds.), Algorithm Engineering and Experimentation. Proceedings, 2001. VIII, 231 pages. 2001.

Vol. 2154: K.G. Larsen, M. Nielsen (Eds.), CONCUR 2001 – Concurrency Theory. Proceedings, 2001. XI, 583 pages. 2001.

Vol. 2157: C. Rouveirol, M. Sebag (Eds.), Inductive Logic Programming. Proceedings, 2001. X, 261 pages. 2001. (Subseries LNAI).

Vol. 2158: D. Shepherd, J. Finney, L. Mathy, N. Race (Eds.), Interactive Distributed Multimedia Systems. Proceedings, 2001. XIII, 258 pages. 2001.

Vol. 2159: J. Kelemen, P. Sosík (Eds.), Advances in Artificial Life. Proceedings, 2001. XIX, 724 pages. 2001. (Subseries LNAI).

Vol. 2161: F. Meyer auf der Heide (Ed.), Algorithms – ESA 2001. Proceedings, 2001. XII, 538 pages. 2001.

Vol. 2162: Ç. K. Koç, D. Naccache, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2001. Proceedings, 2001. XIV, 411 pages. 2001.

Vol. 2164: S. Pierre, R. Glitho (Eds.), Mobile Agents for Telecommunication Applications. Proceedings, 2001. XI, 292 pages. 2001.

Vol. 2165: L. de Alfaro, S. Gilmore (Eds.), Process Algebra and Probabilistic Methods. Proceedings, 2001. XII, 217 pages. 2001.

Vol. 2166: V. Matoušek, P. Mautner, R. Mouček, K. Taušer (Eds.), Text, Speech and Dialogue. Proceedings, 2001. XIII, 452 pages. 2001. (Subseries LNAI).

Vol. 2170: S. Palazzo (Ed.), Evolutionary Trends of the Internet. Proceedings, 2001. XIII, 722 pages. 2001.

Vol. 2172: C. Batini, F. Giunchiglia, P. Giorgini, M. Mecella (Eds.), Cooperative Information Systems. Proceedings, 2001. XI, 450 pages. 2001.

Vol. 2176: K.-D. Althoff, R.L. Feldmann, W. Müller (Eds.), Advances in Learning Software Organizations. Proceedings, 2001. XI, 241 pages. 2001.

Vol. 2177: G. Butler, S. Jarzabek (Eds.), Generative and Component-Based Software Engineering. Proceedings, 2001. X, 203 pages. 2001.

Vol. 2181: C. Y. Westort (Eds.), Digital Earth Moving. Proceedings, 2001. XII, 117 pages. 2001.

Vol. 2182: M. Klusch, F. Zambonelli (Eds.), Cooperative Information Agents V. Proceedings, 2001. XII, 288 pages. 2001. (Subseries LNAI).

Vol. 2184: M. Tucci (Ed.), Multimedia Databases and Image Communication. Proceedings, 2001. X, 225 pages. 2001.

Vol. 2186: J. Bosch (Ed.), Generative and Component-Based Software Engineering. Proceedings, 2001. VIII, 177 pages. 2001.

Vol. 2188: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. Proceedings, 2001. XI, 382 pages. 2001.

Vol. 2189: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis. Proceedings, 2001. XII, 384 pages. 2001.

Vol. 2190: A. de Antonio, R. Aylett, D. Ballin (Eds.), Intelligent Virtual Agents. Proceedings, 2001. VIII, 245 pages. 2001. (Subseries LNAI).

Vol. 2191: B. Radig, S. Florczyk (Eds.), Pattern Recognition. Proceedings, 2001. XVI, 452 pages. 2001.

Vol. 2193: F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.), Technologies for E-Services. Proceedings, 2001. X, 213 pages. 2001.

Table of Contents

Keynote Address: Security Protocols and the Swiss Army Knife (Transcript of Discussion)

Roger Needham

Microsoft Research

Keynote address is an unnecessarily grand term for what I'm about to do, and I don't particularly guarantee to go on for long — leaving more time for more interesting things. The original theme that was stated for this workshop was broadening the horizons for security protocols and (as is traditional) nobody has said anything about that topic whatever in their papers, so I thought I would see if I could say something about it now so that the theme is not totally overlooked.

Twenty years ago, or thereabouts, there was a certain amount of discussion as to what authentication protocols should actually do. You can either take a very minimalist view which says that the person you thought you were talking to was around recently, a slightly higher view that says the person you thought you were talking to was around recently and if he's still around you've got a shared secret, or you could integrate the authentication stuff into the communication protocol you were going to use in a very complete kind of way. There was a reasonable consensus you shouldn't do the latter. The practical reason put forward for this was quite simple: if your communication does not work you would like to know whether it's the physical communication that has failed or whether somebody is interfering with you, because it's a different expert you call in the two cases.

I remember having this discussion in the late 70's at Xerox PARC, because if you're signing the blocks of a message you might well say, why do we need a communications checksum as well? The answer is, because you want to know what went wrong. It was also the case — quite notoriously at that time — that security protocols were exceptionally difficult to design correctly, and I suspect as a consequence of that people felt well if we've got it right for heavens sake let's not do another one because we'll probably do it wrong.

So the things were regarded as not doing very much, you're very lucky to have got them right if you have got them right, and you tended to say here is a tool we will just use it for everything. It could be that that was the right thing to do, it could be that it *is* the right thing to do, but it also could be that because there has been some advance in knowledge over this period it becomes much more reasonable to design the security protocols fairly freely. If you've got a particular application, design a security protocol that's reasonably tailored to that application, because if the last twenty years of work has not been completely thrown away, we ought to be better at doing it now than we were then.

I don't know whether this is a sensible view or not, but if it is, or can be made to become so, I think one could assert that life will become in some ways, rather more comfortable. Perhaps what we ought to do is try to look where the edges

B. Christianson et al. (Eds.): Security Protocols, LNCS 2133, pp. 1–4, 2001.

of security protocols ought to be. You can state this very briefly in the words of Butler Lampson: authentication is knowing where something came from and confidentiality is knowing where it went to. But that still gives quite a lot of space to play with. I don't have any very good ideas for how one might blur the edges but certainly it's the case that, if the authentication operation is a very large proportion of what you're trying to do, you might as well design a protocol for the whole application with the authentication in it and not separate things out. It's a bit reminiscent of discussions about layered design in communication protocols: that layered design may be a good thing, but layered implementation is foolish.

We've tended to commit the analogous sin with security protocols rather a lot, basically I would claim because we're nervous about whether we'll get it right, and we ought to have the tools available to us now to make it easier to get it right. I think that's basically all I want to say.

John Ioannidis: I've been maintaining for a while, in the context of layered design and layered protocol definitions, the attitude that for security protocols there shouldn't necessarily be a single security protocol or single "layer" where we should put security (despite my involvement in IPSec and things like that), but that for every layer into the protocol cake (conceptual layer not implementation layer) either it should be securing itself or there should be an equivalent security version and that these security versions talk to each other. So just because we have a secure network layer doesn't mean that anything above it should be oblivious to the security on the security product in the network layer. Conversely we shouldn't have 15 different security protocols each talking to the one above and to the one below without knowing what each other does.

Reply: I'm sure that's right. I think in communication protocols in general, pretending you didn't know what the neighbouring layers did has been a problem that has plagued us for a while.

Matt Blaze: Steve Bellovin, quoting someone recently, I don't remember who, pointed out that in regard to layering, we've invented this religion but we've become fundamentalists, and I think I'd subscribe to that quip with respect to communications protocols. With security protocols I'm less sure, because the way in which layers above and layers below can tend to change out from under you, is particularly acute with security protocols. For a trivial example, let's imagine designing a security protocol with the requirement that only say 2^{20} or 2^{32} bits of traffic should be allowed to go through without re-keying. The designer of the protocol knows that otherwise he'll get various types of replay attacks, or wrap around on identifiers, or what have you. The designer of the protocol knows that the application won't ever do that because it does messaging or what have you. And then suddenly someone uses precisely the same framework to send streaming video and everything falls apart in ways that are essentially invisible. I think we see that happening fairly often. Designing security protocols to have very well defined layers that have very well defined requirements, is a simple way to avoid this, and some fundamentalism might be in order there.

Reply: Yes, there's certainly a tendency among people who think about security to be fundamentalists for a variety of reasons which would be a separate discussion, but I suppose it's a question of what it costs to do it "properly". If the costs of doing it in the fundamentalist way are extremely low, I suppose most people would say, well do that. If it significantly added to the cost of the transaction then we might say, do it the other way.

Larry Paulson: I'm not entirely sure what you mean by layering, but certainly the methods I know for looking at protocols work when you have the entire protocol in front of you and don't work if you suddenly imagine you are replacing some atomic operation, say if you replace a primitive encryption operation by a one-time pad. I'm not sure that I should know how to reason about systems of protocols unless they are all being analysed at the same time.

Virgil Gligor: The problem that Matt has been pointing out is not a problem peculiar to security. Actually it occurs in other areas of systems design. For example, the major advent of database management systems in the early to mid 70's showed that many of the concurrency- control and recovery protocols in operating systems didn't really do anything for database systems. In fact they got in the way, so the database system designers had to invent those mechanisms and protocols for their own application. The lesson there is, I believe, that we should not hard-wire into lower layers, mechanisms that we could not avoid later in the higher layers.

John Ioannidis: I really like that example, because there is a direct translation of it today. The Voice-over-IP people are rolling their own security protocols for the transfer of data because IPSec is too general. Generality has a price. It may be that generality is actually what we want and we are likely to get sort of an economy of scale, but in other realms generality has a price. The example of operating systems standing in the way of databases is actually a very good one.

Reply: Yes, "the price of generality is unwanted decisions".

Ross Anderson: But if one looks at how this works in practice with banking encryption devices, we have a useful and concrete model. Firstly you've got a lot of devices out there in the field that have a command syntax of say fifty transactions, there are a couple of hundred verbs, or whatever, and it starts off by doing things like encrypt PIN, calculate MAC, and so on. Then that becomes an interface on which everybody has to build, because it's what's sold and what is approved. What happens then is that somebody goes and builds a protocol to talk ISO-8583 which handles banking transactions, and then other people implement that on top of other pieces of hardware, then other people come along and build other protocols which have to be supported on the banking hardware, and so you end up extending the banking hardware so that it will support both ISO-8583 and other stuff. So you've got this crab-wise development, up and down, up and down, and the big risk is that you end up with something so complex that you don't understand it, and you end up with trouble. So this is the real process management problem.

Audience: Seen to a hammer, everything is a nail.

Ross Anderson: It's more than that, I mean you start off with the hammer, then you invent screws and hit them in with the hammer.

John Ioannidis: Or with your wrench.

Reply: And then you end up with both implements highly unoptimised for the purpose.

John Ioannidis: Why use a hammer to pound a screw when you have a wrench?

Reply: Yes, what you're talking about is the inevitable evolution of the Swiss army knife [laughter]; and the analogue of the Swiss army knife for security protocols is even more alarming.

Tuomas Aura: One difference between protocols for the traditional data transfer and for these new applications like voice over IP, or other voice communications and video, is that their concept of integrity is different. Traditionally you would think that message integrity is protecting you so that not a single bit has been corrupted, but for voice you do want to allow bit errors. You don't want to correct them all otherwise a mobile phone would be doing error correction all the time. That is one reason why these new applications need new protocols on them.

Reply: Yes, that's an interesting point, not one that's usually made. I have tended to think that way in connection with such things as encrypting video, but it's certainly true of voice as well.

William Harbison: I'd just like to make a couple of observations about how things actually *have* changed over the years.

When we started this workshop, we deliberately chose the title "Security Protocols" rather than "Cryptographic Protocols". This was considered rather radical at the time, indeed we were often told that one could not have security protocols that did not involve cryptography. I think that there are very few people who would hold that as an absolute article of faith these days, indeed I think many of us know situations where encrypting messages can in fact reduce the security of the system rather than enhance it.

The second observation is that one sees, particularly in certain areas, protocols being designed which are very clever, very intricate, and which come (in the paper which describes them) with an associated set of assumptions, and which are then implemented in a totally different place where a totally different set of assumptions actually apply, and it's the protocol that's blamed rather than the implementation. In fact what has happened is that people have taken a solution from one framework and placed it in another, without understanding the difference between them.

Reply: I'm sure that's right. One of the serious pleas to anybody who publishes in this area is to say what you have assumed, and say it in bigger type than the rest of the paper.

Mergers and Principals

Dieter Gollmann

Microsoft Research
Cambridge, United Kingdom
diego@microsoft.com

Abstract. The term 'principal' has roots both in computer security and in communications security. We will show that in those two areas principals serve quite different purposes. We also note that the term principal is overloaded in computer security and propose a separation into three different aspects: origin of message, access control rule, and accountable entity. Furthermore, we will defend the merits of extensional security specifications and show that it is not fruitful to expect that security mechanisms can only have one 'correct' interpretation.

1 Introduction

The term 'principal' figures prominently in discussions about distributed system security, in particular in the context of authentication. Like 'authentication', the meaning of 'principal' is obvious until it is subjected to closer scrutiny. Although it would be desirable to find one accepted – or acceptable – definition of principals, we will have to settle for elaborating and separating the different usages of this term. As we will try to demonstrate, some of the ambigiuties in terminology result from different historic roots. Principals were used both in computer security (distributed system security) and in communications security. Computer security and communications security supposedly merged about a decade ago, and the two areas definitely used the same language to discuss security concerns. We will examine the effects of this merger on our understanding of principals.

In this paper, we will conduct two case studies. The first case study explores the historic roots of the term 'principal', showing that principals serve quite different purposes in computer security and communications security respectively. The second case study deals with formal semantics for SDSI name resolution. Again, it will become apparent that computer security and communications security can take contrary views of the effects of the same operation. In the light of these observations, we suggest that it is time to contemplate some kind of de-merger, i.e. to properly separate concerns of computer security and communications security and to guard ourselves against letting ideas from one area misguide our understanding of concepts in the other.

B. Christianson et al. (Eds.): Security Protocols, LNCS 2133, pp. 5–13, 2001.

2 Principals

We will trace the history of the term 'principal' in attempt to answer questions like: What is a principal? Where do principals come from? What purpose do principals serve? Is there a future for principals?

2.1 Communicating Principals

A first set of quotes is collected from publications that are concerned with authentication in network communications. Publications on Kerberos figure prominently in our selection.

> Principal: A uniquely named client or server that participates in a network communication [15].
>
> A *principal* is the basic entity which participates in network authentication exchanges. A principal usually represents a *user* or the *instantiation of a network service* on a particular host [13].
>
> After authentication, two *principals (people, computers, services)* should be entitled to believe that they are communicating with each other and not with intruders [4].
>
> The fundamental purpose of authentication is to enable *"principals"* to identify each other in a way that allows them to communicate, with confidence that the communication originates with one principal and is destined for the other. The principals we are considering include *people, machines, organizations and network resources such as printers, databases or file systems* [2].
>
> In distributed computing systems and similar networks of computers, it is necessary to have procedures by which various pairs of *principals (people, computers, services)* satisfy themselves mutually about each other's identity [3].

In summary, principals are entities that communicate with each other and who can be recognized (authenticated) in a conversation. Principals are not necessarily human users. They can equally be a machine or a network service. In communication security, principals are peer entities that can run authentication protocols. They are the source of messages, but the content of these messages is irrelevant for our current considerations.

We note in passing that there exists an interpretation of entity authentication, given in International Standard ISO/IEC 9798-1, where the authenticated principal only has to show that it is alive, i.e. active during the run of the authentication protocol. This definition is not concerned with establishing secure conversations.

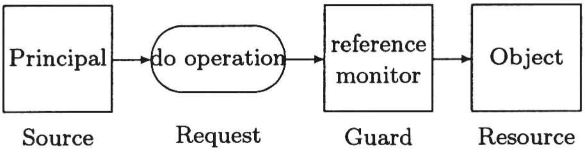

Fig. 1. A model for access control

2.2 Principals in Access Control

In computer security, we meet a scenario where messages are not being sent between equal partners, but more likely between a client and a server. Furthermore, messages are access requests and servers refer to the content of a message when making access control decisions.

> All authentication is on behalf of principals. Principals can act in two capacities, *claimant* or *verifier*. SPX recognizes the following two types of principals, *users (normally people with accounts)* and *servers* [18].

> A *principal* is an entity that can be *granted access* to objects or can make statements affecting access control decisions [7].

> Subjects operate on behalf of *human users* we call *principals*, and access is based on the principal's name bound to the subject in some unforgeable manner at authentication time. Because access control structures identify *principals*, it is important that principal names be globally unique, human-readable and memorable, *easily and reliably associated with known people* [6].

The last quote views principals in a fashion rather different from communications security. Principals are entries in access control structures. To make access control manageable they are closely associated with human users. Principals are sending messages only metaphorically, the actual work is done by subjects operating on their behalf. The following quote sums up the role of principals in access control (see also Figure 1).

> If s is a statement *authentication* answers the question "Who said s?" with a principal. Thus principals make statements; this is what they are for. Likewise, if o is an object *authorisation* answers the question "Who is trusted to access o?" with a principal [14].

Usually the access control structure is attached to the object as an *access control list (ACL)*. For each operation, the ACL specifies a set of authorized principals. To support a wider range of access control policies, the concept of 'principal' is further elaborated and [14] distinguishes between *simple* and *compound* principals. Simple principals are: