Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 64
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 64
Espoo 2000 HUT-TCS-A64

AUTHORIZATION AND AVAILABILITY - ASPECTS OF OPEN NETWORK SECURITY

Tuomas Aura

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 64
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 64
Espoo 2000 HUT-TCS-A64

AUTHORIZATION AND AVAILABILITY - ASPECTS OF OPEN NETWORK SECURITY

Tuomas Aura

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering, for public examination and debate at the meeting room of the Council of Helsinki University of Technology (HUT main building, Espoo, Finland) on the 17th of November, 2000, at 12 noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory for Theoretical Computer Science

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietojenkäsittelyteorian laboratorio

Distribution:
Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O.Box 5400
FIN-02015 HUT
Tel. +358-0-451 1
Fax. +358-0-451 3369
E-mail: lab@tcs.hut.fi

ISBN 951-22-5218-X
ISSN 1457-7615

Otamedia Oy
Espoo, Finland 2000

ABSTRACT: The world is becoming increasingly dependent on secure, reliable access to services on the Internet and in other open communications networks. Since the administration and authority on these networks are completely distributed, it is not possible to set or enforce global security policies. While security and confidentiality of data are still significant concerns, access control and resistance to denial-of-service (DOS) attacks have become at least as significant security goals. Traditional methods for access-right management and resource allocation, which were defined for centrally administered systems, are not applicable on the open networks. Consequently, new techniques for access control and DOS prevention are needed.

This dissertation addresses several aspects of the security of open, distributed systems: decentralized access control, design of key-establishment protocols, and denial-of-service resistance. We suggest technical solutions both for extending the scope of applications that can securely be run on the networks and for improving the reliability of the underlying infrastructure for all applications.

We define a formal model of key-oriented access control and use this model to develop algorithms for access-control decisions from a certificate database. We survey privacy protection in public-key infrastructures, introduce a new kind of threshold certificate, and present novel certificate-based solutions for access control between mutually distrusting software packages on intelligent-network routers and for software license management with smartcards. We also describe novel design principles for cryptographic protocols to improve their robustness against common replay attacks at a low cost and to protect on-line services against denial-of-service attacks that attempt to exhaust server memory and computational resources. Additionally, we develop a method for analyzing the vulnerability of network topologies to denial of service by the destruction of communications links.

Throughout, the emphasis is on security issues critical for the commercial and private use of the Internet and other open communications systems where mutually distrusting entities must share resources and co-operate.

KEYWORDS: open network security, authorization certificates, cryptographic protocols, denial of service

CONTENTS

PREFACE

This dissertation is the result of studies and research at the Laboratory for Theoretical Computer Science of Helsinki University of Technology (HUT) from 1996 to 2000. I am grateful to my supervisor, Professor Leo Ojala, for providing the excellent environment for theoretical research, and to all the people at the laboratory for creating the demanding and encouraging atmosphere. In particular, I want to acknowledge the frequent advice and enthusiastic spirit of Professor Ilkka Niemelä.

When starting licentiate and doctoral studies on network security, my background was in formal methods and I knew little about this topic. Discussions with Pekka Nikander and Professor Arto Karila from HUT Telecommunications software laboratory got me interested in security. They and many other people from their research group have since been an endless source of insights into computer networking and telecommunications.

During the year 1998–1999, I had the opportunity to work at the University of California, Davis, Computer Security Laboratory with Professors Matt Bishop and Carl Levitt and their research group. From the frequent discussions with them, I learned a lot about applied security, classic security literature, and about the computer security research process.

The co-authors Matt Bishop, Carl Ellison, Dieter Gollmann, Jussipekka Leiwo, Pekka Nikander, and Dean Sniegowski naturally played major roles in creating the publications included in this dissertation. It has been an invaluable part of my learning and research to be able to exchange ideas with such experienced and talented people.

The insightful comments of the pre-examiners N. Asokan and Paul Syverson were also of great value in preparing the final version of the thesis.

The work was funded by Helsinki Graduate School in Computer Science and Engineering (HeCSE) and by the Academy of Finland (projects #44806, and #47754). It has also been generously supported by the Finnish Cultural Fund, Finnish Foundation for Technology, Foundation for Financial Aid at HUT, Ella and Georg Ehrnroot's fund, Emil Aaltonen foundation, Alfred Kordelin's foundation, Telecom Finland research and educational fund, and Kaupallisten ja teknisten tieteiden tukisäätiö foundation. I gratefully acknowledge the importance of this support as it made full-time studies and frequent international contacts possible.

Yulian and our little son, Sampo, made this work much more enjoyable.

Otaniemi, November 2000

Tuomas Aura

LIST OF PUBLICATIONS

The dissertation consists of 9 publications that are listed below. Publications [**P1**]–[**P5**] are on the management of access rights with certificates, [**P6**] on protocols for creating secure connections between a client and a server, and [**P7**]–[**P9**] on denial-of-service (DOS) issues.

[**P1**] Tuomas Aura, Distributed access-rights management with delegation certificates, in J. Vitek and C. Jensen (Eds.): *Secure Internet Programming – Security Issues for Distributed and Mobile Objects*, volume 1603 of Lecture Notes in Computer Science, pages 211–235, Springer 1999.

[**P2**] Tuomas Aura, On the structure of delegation networks, in *Proceedings of 11th IEEE Computer Security Foundations Workshop*, Rockport, MA USA, June 1998, pages 14–26, IEEE Computer Society Press 1998.

[**P3**] Tuomas Aura, Fast access control decisions from delegation certificate databases, in Colin Boyd and Ed Dawson (Eds.): *Proceedings of Information Security and Privacy, 3rd Australasian Conference (ACISP'98)*, Brisbane, Australia, July 1998, volume 1438 of Lecture Notes in Computer Science, pages 284–295, Springer 1998.

[**P4**] Tuomas Aura and Carl Ellison, Privacy and accountability in certificate systems, Research Report A61, Helsinki University of Technology, Laboratory for Theoretical Computer Science, editor Leo Ojala, Espoo, Finland, April 2000.

[**P5**] Tuomas Aura and Dieter Gollmann, Software license management with smart cards, in *Proceedings of USENIX Workshop on Smartcard Technology*, Chicago, IL USA, May 1999, pages 75–85, USENIX Association 1999.

[**P6**] Tuomas Aura, Strategies against replay attacks, in *Proceedings of 10th IEEE Computer Security Foundations Workshop*, Rockport, MA USA, June 1997, pages 59–68, IEEE Computer Society Press 1997.

[**P7**] Tuomas Aura and Pekka Nikander, Stateless connections, in Yongfei Han, Tatsuaki Okamoto and Sihan Qing (Eds.): *Proceedings of Information and Communications Security, 1st International Conference (ICICS '97)*, Beijing, China, November 1997, volume 1334 of Lecture Notes in Computer Science, pages 87–97, Springer 1997.

[**P8**] Tuomas Aura and Pekka Nikander and Jussipekka Leiwo, DOS-resistant authentication with client puzzles, to appear in *Proceedings of Security Protocols Workshop 2000*, Cambridge, UK, April 2000, Springer.

[**P9**] Tuomas Aura and Matt Bishop and Dean Sniegowski, Analyzing single-server network inhibition, in *Proceedings of 13th IEEE Computer Security Foundations Workshop*, Cambridge, UK, June 2000, pages 108–117, IEEE Computer Society Press 2000.

The current author is the main author of all the publications. [**P1**]–[**P3**] and [**P6**] were written without co-authors. The survey of privacy problems and solutions in [**P4**] is written by the current author but heavily based on published and unpublished work of the co-author, Carl Ellison. The idea of open threshold certificates is the current author's. In [**P5**], the certificate-based solution for software license management is the current author's. The co-author, Dieter Gollmann, provided the original problem and expertise in the application area. In [**P7**], the idea of statelessly maintained connections is the current author's. Discussions with the co-author, Pekka Nikander, opened eyes to the value of the concept and he provided the insight into the Internet protocols. The basic ideas of [**P8**] evolved in discussions with the co-authors, Pekka Nikander and Jussipekka Leiwo. The final solution is the current author's work. [**P9**] is a result of collective work with the co-authors, Matt Bishop and Dean Sniegowski. The proofs, logic-program representation and final text are by the current author.

1 INTRODUCTION

On the Internet, an unlimited number of mutually distrusting persons and organizations share resources and make services available to each other. The telecommunications world is moving towards the same kind of solutions where competing enterprises co-exist and co-operate in one network. One common feature of these networks is that they are open: anyone may join in as a participant, share the communications channels, and provide or refuse to provide services to others. Another characteristic feature is that the administration of these information systems is highly distributed. There is no single entity that could set access policies or enforce them. Instead, everyone controls access to the resources they own.

From the security point of view, these open and decentralized systems differ significantly from the closed, centrally administered computers and networks for which security mechanisms have traditionally been designed. While confidentiality and integrity of data are still important concerns in today's networks, access control and service availability have become equally critical. Businesses that sell and exchange their services on the network need flexible ways of administering access to their products. The access control mechanisms must adapt to the complex trust relations in the real world and support both old and new business models. Moreover, the continual functioning of the networks has become critical for the day-to-day operation of businesses and governments but the Internet in particular has not been designed to be reliable enough for such critical applications.

In this dissertation, we look at several aspects of network security that arise in a world which increasingly depends on the secure, reliable functioning of the communications networks. We suggest technical solutions both for extending the scope of applications that can securely be run on the networks and for improving the reliability of the underlying infrastructure for all applications. The publications included in this dissertation cover a relatively wide range of topics with the common factor being access control and availability of services in open networks.

The approach is theoretical where formal definitions and proofs have helped to avoid ambiguity and to give evidence about the validity of the new ideas. As is typical for computer science, simulations and experiments serve as further evidence but are unable to accurately predict the usability of the theories in real applications. The ultimate criteria for the success of the research will be the acceptance of the results for applications and for further development by the research community. It has therefore been encouraging to see that some novel ideas from the publications of this dissertation have already been improved on by other researchers, and application developers have at least shown interest in the results.

The dissertation consists of this introduction and nine original publications. In the introduction, we give an of overview the problems and literature that motivated the research, and link the results of this dissertation to the prior work.

1.1 Contributions

The dissertation addresses three critical aspects of security in open, distributed information processing systems: decentralized access right management, security of cryptographic protocols, and protection of servers against denial of service.

Publications [**P1**]–[**P5**] discuss access-right management with signed certificates. The overall goal has been to understand the potential and limitations of key-oriented access control and to extend the range of its applications.

The rest of the publications address the security of service access over an open network. Publication [**P6**] presents new design techniques for robust authentication and key-exchange protocols and [**P7**]–[**P9**] focus on the resistance of protocols and network topologies against denial-of-service attacks.

The main contributions of the dissertation, and of each publication, are the following:

1. A formal model of the delegation of access rights with certificates and an algorithm for access-control decisions that is based on this model.

   Publication [**P2**]

   (a) presents a formal model of delegation where the signature keys and certificates form a graph, in which authorization is characterized by the existence of certain embedded paths and trees,

   (b) shows that the SPKI certificate reduction semantics is sound and complete with respect to the model, and

   (c) introduces a new type of threshold certificate that improves privacy protection and adds flexibility to certificate management.

   Publication [**P3**]

   (a) presents an efficient two-way graph-search algorithm for deciding whether a set of certificates authorizes a given access request.

2. A survey of privacy protection in public-key infrastructures and a new kind of threshold certificate that enhances the privacy of the subjects.

   Publication [**P4**]

   (a) overviews privacy concerns in public-key infrastructures and techniques for improving the privacy of individuals and organizations, and

   (b) explains how individuals can be held accountable for the certificates and access requests they have issued without compromising the privacy protection.

   The techniques of [**P4**] have previously been used for privacy protection and for other purposes in the literature. (The new threshold certificates are from [**P2**].) We show how they work together to protect privacy in a key-oriented PKI.

3. Novel certificate-based solutions for access control between mutually distrusting software packages on intelligent-network routers and for software-license management with smartcards.

   Publication [**P1**]

   (a) overviews SPKI-style delegation certificates,

   (b) gives an example of their use in intelligent networks (IN), and

   (c) shows how threshold certificates and novel conditional certificates can be used to encode common security policies and how they add flexibility to access-right management.

   The overview (a) is based on the SPKI literature while (b) and (c) are original solutions.

   Publication [**P5**]

   (a) shows how access rights, such as software licenses, that are stored on tamper-proof modules can be implemented with delegation certificates, transfered between modules and purchased online, and

   (b) proves the security of the transfer protocol, discusses the assumptions of tamper-proofness and suggests ways of alleviating the risks.

4. Systematic, low-cost design principles for cryptographic protocols that make the protocols robust against common replay attacks.

   Publication [**P6**]

   (a) explains how unique cryptographic functions, such as hash and encryption functions with unique initializers, can be used instead of explicit type tags,

   (b) shows how redundant but security-wise important data in protocol messages can be compressed by hashing it and suggests this as a way of implementing the full-information principle, i.e. including the maximal amount of redundant data in each message, and

   (c) suggests binding cryptographic keys to their intended use by hashing key parameters together with the key.

   Ideas (a) and (b) have not been previously publicly documented although it is likely that similar techniques have been used in applications. (c) is a generalization of a technique used in various protocols.

5. Design techniques that protect on-line services against denial-of-service attacks by resource exhaustion by maintaining connections with stateless servers and by ensuring that the client of an authentication protocol commits its resources before the server does.

   Publication [**P7**]

   (a) describes a secure transformation of a stateful protocol into a stateless one in order to improve its resistance to DOS attacks, and

(b) gives examples of situations and protocols where the server benefits from being stateless for some period.

Publication [**P8**]

(a) describes how the server in an authentication protocol can protect itself against resource exhaustion attacks by asking the clients to solve small cryptographic puzzles before it allocates memory for the protocol state or performs expensive cryptographic computations.

Similar, slightly more complex, client puzzles have previously been used for DOS protection in other applications.

6. A method for analyzing the vulnerability of network topologies to denial of service by the destruction of communications links and nodes.

   Publication [**P9**]

   (a) defines a new type of DOS attack against communication networks where the attacker tries to disconnect as many clients as possible from a single server by removing links and nodes from the network,

   (b) proves that finding optimal attacks is an NP-complete problem, and

   (c) develops a technique for analyzing networks based on logic programs with stable-model semantics.

   Related attacks have been studied in graph-theoretic literature but this one appears to be an independent problem.

Throughout, the emphasis is on security issues critical for the commercial and private use of the Internet and other open communications systems where mutually distrusting entities must share resources and co-operate.

2 ACCESS-RIGHT MANAGEMENT

> *"What I propose here is a generalized certificate, not tied to identities but in which identity certification is possible when needed."*
>
> *Carl Ellison [46]*

The communications networks are increasingly used for accessing services over the network. Access control is traditionally divided into two stages: authentication and authorization. Authentication in this division means verifying the user identity, and authorization means checking that the user has the right to access the service.

Currently most services authenticate their clients by asking a password. However, the password control is relatively insecure and geared towards human users. When software systems increasingly need to access each other (e.g. over Common Object Request Broker Architecture (CORBA) and Java Remote Method Invocation (RMI)) with minimal human interaction, cryptographic authentication is required. Also, more flexible ways of managing access rights are needed to augment the access control lists (ACL).

Key-oriented access control challenges the traditional two-step view by removing the authentication part and binding access rights directly to the user's public signature key. The rights can be stored and distributed in a completely decentralized manner with authorization and delegation certificates. The author believes that the key-oriented certificates are the most promising approach to automated access-right management in distributed systems.

Publications [**P1**]–[**P5**] relate to discretionary access control with signed certificates. Publication [**P1**] is an introduction to the key-oriented access control. Publication [**P2**] gives a formal semantics to delegation certificates that agrees with the certificate reduction used in the proposed SPKI standard and [**P3**] uses this model to derive an efficient algorithm for making access-control decisions from a certificate database. In [**P1**], [**P4**] and [**P5**], we discuss new applications and privacy enhancements enabled by the key-oriented certificates. Below, we will overview both these new and some more traditional models of access control.

2.1 Classical access-control models

The study of computer security started in the 1970's when multi-user operating systems, where several users may simultaneously store and process data, became popular. In military environments, access to classified data has to be restricted, and in civilian systems, the users often have conflicting interests and privacy concerns. Hence, the operating system has to give the users protection from each other's actions.

Lampson [75] was the first to formally define the basic concepts of protection: the *access control matrix* and its two representations, *access control lists* and *capability lists*. In the former, a list of authorized users is stored for each resource, while in the latter, a list of access rights is stored for each user. These concepts are general enough to still be in wide use today.

Lampson also identified the *confinement problem* [76]: how to prevent a service from wilfully leaking confidential data through legitimate or covert communications channels. This kind of mistrust towards the users and sys-

tem components motivated much of the military-backed research on computer security until late 1980's. The goal has been to design *mandatory access controls (MAC)* that by technical means make impossible all access and communication that is not permitted by a system-wide *security policy*. The mechanisms that provide this total control over user actions are collectively called a *trusted computing base (TCB)* [92].

The early computer security policies were copied directly from the US military organizations. In a *multilevel-secure* operating system, users are assigned a *clearance* and objects a *classification*. The Bell-LaPadula model [21] protects the confidentiality of data by preventing the flow of high-level information to users with lower clearance. The Biba model [23] similarly protects the integrity of high-level data by preventing the flow of information from low to high levels. The types of access considered in these models are reading and writing (or modifying) of files. Several models have been proposed for verifying that a particular system conforms to a given policy [110]. As Harrison, Ruzzo and Ullman [60] showed, the security of a system is undecidable under fairly weak assumptions.

The mandatory control does not leave any policy decisions to the users themselves. However, it is often desirable to let users control access to data created or owned by them. Several modification to the strict MAC have been proposed for this purpose [6]. For example, in *originator-controlled access control (ORGCON)*, those who can read a file are not allowed to make copies that they could later read, modify or distribute to others. In *owner-retained access control (ORAC)* of McCollum et al. [83], the originator of a file retains control over it, and all access must be authorized by the authors.

Although the military-style classification of data has been used in commercial applications (for example, as described by Lipner [78]), the various civilian environments have their own typical policies that differ from the military systems. Two well-known access control policies that build on commercial data processing and accounting practices are the Clark-Wilson model [37] and the *Chinese wall* policy of Brewer and Nash [30]. These policies should be enforced by mandatory controls and thus require a TCB. The latter is a *non-monotonic* policy where conflicts of interest are prevented by disallowing access to files if a conflict arose with other files the user has previously accessed.

Discretionary access control (DAC) [43, 93] differs from mandatory in that the users are trusted to grant access rights to each other. The users may allow or disallow access to resources they control. There are two critical questions related to DAC. First, it is debatable to what extent the access rights can be confined to any set of users. Since the users are trusted to communicate and provide services to each other, it is difficult to prevent them from passing restricted information to each other. For example, a user who is able to read a file on her own workstation can usually make copies and distribute it to others. While many services, like access to dynamic databases, cannot be copied as easily as static files, a user with access to a service may act as a proxy and pass requests from other users to the service. This means that the confinement problem is not limited to confidential data but extends also to access rights for controlled services. Second, even co-operating users may unknowingly distribute data and access rights. This is because careless users may

execute programs that contain *Trojan horses*, hidden features that do things the user has not agreed to. Without mandatory controls, a Trojan horse may circumvent whatever rules the user has voluntarily decided or promised to follow.

Under a single operating system, the mechanisms for discretionary control are the same as those used for enforcing mandatory policies: a reference monitor that allows or disallows access according to ACLs and capability lists, or more typically, user groups and protection bits, which are another representation of the access control matrix. In a distributed environment, new mechanisms are needed.

2.2 Distributed access control

While mandatory access control in a distributed system requires the TCB to be embedded in all the system components so that a global policy can be enforced everywhere, discretionary access control can be implemented entirely without central control. Instead, resources may be controlled locally by the owner or by the network node where they reside. The local control enables many new ideas in the management of access rights and identity information. In particular, the capability lists, which are traditionally stored at the location of the service, may be distributed to the clients as cryptographically protected *credentials*.

Comprehensive credential-based systems for distributed access right management are described, for example, by Gong [54] and Bull et al. [31]. The latter system is worth noting because it provides much of the functionality of advanced public-key infrastructures with only shared-secret mechanisms (although public-key signatures are also allowed).

In Bull's system, a server itself, or its trusted agent, authorizes clients to use the services by issuing access certificates (i.e. credentials). It signs them with a secret-key algorithm using keys that are known only to the server and its trusted agents. This way, the servers exercise local control over the services that they provide. Moreover, the clients may *delegate* the access rights to each other by adding signatures on the access certificate if one client needs another one to issue a request on its behalf.

This may result in a *chain of delegation*. The chain can be verified by the server because it begins from the server itself. This kind of situation where an authorization is verified by the entity from which it originates is called an *authorization loop*. As its first step, the loop usually contains an entry in the server's ACL indicating that some agent is trusted to delegate rights to a certain service.

A much-studied problem in a distributed system is the *revocation* of access rights. In centrally controlled systems, it is possible to revoke access rights immediately. In a distributed environment, the non-monotonic changes to access rights are more difficult to enforce and, due to delays in the propagation of the revocation information, a margin of error always remains. There are arguments both for on-line databases or periodically distributed lists of revoked credentials and for short-lived, frequently refreshed certificates without a possibility for revocation.

2.3 Public-key certificates and trust metrics

The main problem with the distributed access-control systems based on symmetric cryptography is key management. The identity and access credentials and the associated secret keys must be distributed on-line. Moreover, a separate secret key is needed for each authority-server and client-server pair. Public-key encryption and signature algorithms [41] simplify key management significantly because only the public keys need to be distributed, and only one or a few keys are needed per entity.

Nevertheless, key management remains the greatest technical challenge even in public-key cryptography. The solutions vary from the anarchic PGP web of trust [120] to the X.509 *public-key infrastructure (PKI)* [34] with a hierarchy of *certification authorities (CA)*. Most, however, rely on *identity certificates*, originally introduced by Kohnfelder [72], that bind a public key to the name of the owner of the corresponding private key and are signed by a trusted entity.

Inside a single organization where the administrators are trusted by everyone, it is fairly easy to set up a PKI according to the organizational hierarchy. When a server wants to verify the public key of a client, it can find a *certificate chain* from a CA it knows, through higher level authorities, to one that has certified the client directly.

Attempts have been made to extend the hierarchical CA model to inter-domain authentication but none has had much success beyond occasional bilateral cross-certification between two hierarchies. The reason is that one must trust a CA with all the data and access rights that are communicated with the help of its certificates. Moreover, in a chain of identity certificates, one must transitively trust each authority to recommend the next one as equally reliable. Between and outside organizations, where the absolute trust is lacking, the semantics of a certificate chain becomes unclear. Several probability-based metrics for the reliability of certificate chains have been proposed, e.g. by Beth et al. [22] and by Maurer and Kohlas [82, 70]. Independent paths can be combined for increased assurance with the techniques of Reiter and Stubblebine [101, 102]. However, all such metrics depend on estimates of the reliability of the individual certifiers, which is difficult to measure.

There are several other issues that arise from the use of identity certificates [51]. One major problem is that the names of the certified individuals must be globally unique and unambiguously recognizable by all those who rely on the certificates. It has proven difficult to create reliable global naming schemes. Furthermore, a PKI only maps public keys to names and leaves the more important questions of managing access rights to other mechanisms.

2.4 Key-oriented access control

New *key-oriented* access control methods address the problems of traditional PKIs in two ways. First, certificates are used to bind keys directly to attributes and authorizations (instead of names) and, where names are used, they are defined and interpreted relative to the entities that use them (as opposed to being globally known and unique).

Attribute certificates extend the semantics of identity certificates so that, instead of only binding names and keys to each other, arbitrary properties may be bound to a name or to a key. In strictly key-oriented systems, attribute certificates are issued only to the public keys. (In identity-oriented systems, the attributes are bound to names, which in turn are bound to public keys with identity certificates.) Simple examples of attributes are clearance levels, email addresses and dates of birth. *Authorization and delegation certificates* are public-key versions of access credentials. They are used to bind access rights to public keys and to distribute them from key to key. Anyone may issue these certificates. The attributes and authorizations are considered valid only by those who trust the issuer. Issuing the certificates directly to the public keys has the advantage that no identity certificates or certification authorities are needed. This makes the system less centralized and more reliable.

The attribute and authorization certificates are used in the same way as all credentials. The user presents the certificates to the server with its access request. The server verifies their integrity with the public key, which it must know. It then consults the *local security policy* to determine whether the credentials entitle the client to the service. The local policy may impose restrictions, such as limitations on delegation, that override the contents of the certificates.

There are two major proposals for a key-oriented PKI: the KeyNote trust management system by Blaze et al. [26, 25] and the SPKI certificates by Ellison et al. [47, 50, 49, 48]. KeyNote is based on PolicyMaker [27] where the certificates can be arbitrary programs. KeyNote restricts the expressive power of the certificates to essentially the same level as SPKI, so that checking the compliance of an access request is computationally easier. While PolicyMaker is mostly a research tool, SPKI and KeyNote are more geared towards practical applications. Although SPKI certificates were the basis for the publications [**P1**]–[**P5**], many of the results apply to KeyNote certificates as well.

One common feature of the new PKIs is that they obscure the boundary between policy and credentials. The security policies that are traditionally maintained locally at the servers can now be encoded into the certificates and distributed to the clients. This is particularly clear in KeyNote where the local policies have the same syntax as the credentials, and any policy statement can be signed into a credential. In a distributed system, if the server is not always on-line, or if the same credentials are accepted by many servers, it may be easier to distribute the policies as certificates than to reconfigure the servers. The policy certificates add flexibility: policies can be changed on demand and personalized for each client. On the other hand, if the expressive power of the certificates is increased so that arbitrarily complex policies can be encoded in them (as in PolicyMaker), the certificates become too complex to interpret. Hence, static and complex policies should be implemented at the server rather than encoded in the certificates.

SPKI defines versatile certificates with an elegant semantics that is built around the delegation of access rights and certificate reduction. The delegation certificates can form chains where access rights are granted from a public key to a public key. Every certificate must explicitly state the rights it gives. Hence, SPKI certificates are transitive but the purposes for which the

chain is valid can be limited in every certificate. This way, delegation solves the problem of transitive trust by making the trust less absolute. The first part of [**P1**] explains in detail how delegation certificates are used to distribute access rights. The second part suggests some extensions to the certificate semantics so that more complex policies, such as requiring the approval of a third party, can be expressed in the certificates.

Rivest and Lampson redefined the concept of a name in a novel way in the SDSI [103] infrastructure. The SDSI names can be defined locally by anyone and they do not need to be unique. (The names can, in fact, be seen as groups.) This way, a name becomes just another kind of an attribute that is always interpreted relative to the issuer of the name certificate. Moreover, the SDSI names may point to each other and have more than one level of relative references. All names are eventually resolved into one or more public keys. The original SDSI was a comprehensive public-key infrastructure and largely based on its powerful name system. The SDSI names have since been integrated into SPKI.

Although the key-oriented access-control systems mentioned here all have fairly rigorous definitions, it is still worth trying to formulate an abstract semantics that would separate the essential ideas from implementation details. Publication [**P2**] views the certificate chains and threshold certificates as a generalized graph structure where valid authorizations correspond to embedded trees. This definition is shown to agree with the SPKI reduction semantics. Other, logic-based definitions have been given by Li et al. [77] and by Kohlas and Maurer [71]. The logic by Lampson, Abadi and others [74, 3], which formed the basis for security in the TAOS operating system [117], is versatile enough to be used for reasoning about key-oriented access control systems even though it predates them. Logics to clarify the meaning of linked SDSI names have been presented by Abadi and by Halpern and van der Meyden [2, 58].

2.5 Applying delegation certificates

The delegation certificates enable new applications, many of which may still be undiscovered. We explored the use of certificates for two fundamentally different applications.

In [**P1**], we found novel ways of managing co-operation between code modules from mutually distrusting service providers on intelligent-network routers. The service providers may delegate access rights to each other. They may also require quality certification from each other before allowing access. To minimize synchronization between the participants, these kinds of conditions may be encoded into the certificates, either by using the SPKI threshold certificates in creative ways or by extending the certificate semantics to support such conditions. In either case, the conditions are satisfied by providing further certificates from appropriate authorities. The possibility of encoding client-specific policies into the certificates adds a lot of flexibility in comparison to reconfiguring the servers for each new client and the associated policy details.

Publication [**P5**] presents novel techniques for software license management. The aim of this paper is to explore the limits of using hardware to-

kens for copy protection assuming that the code cannot be executed on a trusted processor and that there are no unique identifiers for the workstations. We bind the licenses to smartcards and use delegation for transferring them between the cards. Admittedly, the copy-protection in our system depends on some hard-to-implement assumptions, most significantly on the *tamper-resistance* of the operating system code and the smart card hardware, and it is possible for several workstations to share a single license on-line over a network by tapping between the card reader and the operating system. Nevertheless, we are able to improve the flexibility of existing token-based copy protection without further weakening its security. One notable idea in the paper is that the license transfer protocol should be designed so that the failure of old tamper-resistant tokens does not automatically compromise the security of new ones.

Another recent application of SPKI certificates is in the authorization of Mobile IP users who roam in foreign wireless networks. (We describe a simplified version of the system documented by Weckström [116].) Both the users and the wireless network owners sign contracts with brokers, such as ISPs or telecoms. The brokers sign deals with each other and, if there is a chain of brokers from the network to the user, the user is allowed to roam. For each contract in the chain, an SPKI certificate is signed. The user device uses the chain of certificates to prove its right to access the Internet through the foreign network.

In [**P4**], we explain how the key-oriented certificates can significantly enhance the privacy of the users and organizations. Binding the rights to keys instead of to user identities, creation of new keys on demand, and reduction of certificate chains not only adds flexibility to access-right management; these techniques can also be used to protect the privacy of individuals and confidential information about business relations. Although they do not guarantee anonymity because the communication can always be traced, they can greatly reduce the amount of confidential information going around in the certificates. In the Mobile IP example, the user privacy is partially protected because the foreign network does not learn the name of the user. It suffices for the foreign network to know that the user is authorized to use its services. In this case, however, further protection of the user privacy would require changes to the lower level protocol as most Mobile IP users can be identified by their permanent home address. If the address is not permanent, it makes sense for the user to change signature keys periodically.

To hold individuals responsible for their actions and commitments, we advocate the use of *identity escrow*, introduced by Kilian and Petrank [69], which means that the names of the key owners are stored by a trusted third party that reveals them only under some well-defined circumstances. The escrow requirement can be implemented as a condition written on the certificates. These conditions are validated by providing other certificates issued by escrow agencies. Since such conditions are helpful in all the applications we have encountered, it would be a good idea to define general-purpose conditions as a part of the standard certificate syntax. A more expressive semantics for the conditions is a topic for future work.

The generality of the key-oriented certificates, which makes it possible to apply them to almost any discretionary access-control problem, is also the

greatest open question for their future. Every application has its own types of attributes, access rights, and policies for interpreting them. Before the presentation of attributes and access rights has been standardized for an application, the use of the new distributed access control systems is limited to single organizations with a central administration that establishes the rules for interpreting the certificates.

Another open problem, largely independent of the meaning of the certificates, is how the certificates will be stored and distributed to those who need them. If several certificates from different issuers are needed for accessing a service, there is much room for optimizing the paths through which these pieces of data travel. The hierarchical X.509 certificates were originally supposed to be stored in a corresponding hierarchy of directory services, but no such directories are widely available.

One way to manage authorization certificates is to accumulate certificate chains along the path of delegation. As suggested in [50], each key holder that is granted a right should receive a complete sequence of certificates and, upon further delegation, append the new certificate to it. Although convenient in many applications, sticking strictly to this approach has some disadvantages. First, the certificates must be issued and refreshed in a fixed order and they cannot be creatively combined into new chains by the subjects. This undermines the flexibility that authorization certificates provide, e.g. in comparison to cascaded authentication [111]. For example, the Mobile IP user should be able to replace certificates with fresh ones when roaming agreements are extended and to add new certificates to his personal certificate database when his broker signs roaming agreements with new foreign networks. Second, reimplementation of the certificate storage and discovery in each application program should be avoided by creating general-purpose certificate databases. But storing application-specific chains makes it more difficult to implement such databases efficiently. Hence, we believe that algorithms for discovering certificate chains and verifying authorizations from large certificate databases are needed.

Publication [**P3**] presents an efficient algorithm for searching certificate chains in locally stored certificate databases. The idea is that, in order to find a chain between two keys, we search both forward from the first key and backward from the second one. If the searches meet at some intermediate key, a complete chain has been found. Since threshold certificates make a forward search prohibitively expensive, the most practical strategy is to combine a forward search to a small maximum depth or until a threshold certificate is met with a backward search to find one of the same keys. The paper presents a breadth-first backward search algorithm that handles threshold certificates well. The two-way search algorithm was used by Nikander and Viljanen [98], who suggest storing SPKI certificates in the Internet domain name servers (DNS) that form perhaps the most global directory currently available. In the case of Mobile IP, the brokers may eventually form chains of more than one or two steps, and our algorithm might be useful for finding such chains in public directories where the roaming agreements may be stored.

Another algorithm for certificate-chain discovery was presented by Elien [45] (published after [11], in which our algorithm first appeared). It supports SDSI names and works by generating a finite closure set of reduced certifi-

cates, thus computing all valid authorizations at the same time. In comparison, our algorithm handles only authorization certificates and is optimized for verifying a single authorization relation between two given keys.

3 KEY-ESTABLISHMENT PROTOCOLS

> *"The cryptanalysis of protocols is essentially formalized paranoia, since it depends on suspecting everything."*
>
> *Gustavus Simmons [109]*

After the access credentials have been issued and distributed, the client will establish a secure connection to access the server. For most purposes, a secure connection means that the communicating parties possess shared secrets, *session keys*, which they use for authenticating and encrypting the data traffic between them. The step from a credential issued to the name or public key of the client to a shared secret key is more complicated than it may first appear. This process is variably called *authentication* (the client and server verify each other's identities or authorizations), *key distribution* (the session key is decided and distributed by one party) or *key agreement* (all parties contribute equally to the value of the key) depending on the exact goals of the protocol. The protocols for opening authenticated connections have been widely researched in computer security and we can give only a brief overview of the issues involved. Publication [P6] contributes to protocol security by presenting new design techniques that help to avoid mistakes caused by protocol optimization. Protocols that follow the principles may also be easier to analyze and prove correct.

3.1 Protocol failures

The academic study of key-establishment protocols began when Needham and Schroeder [95] in 1978 proposed two protocols for establishing a shared secret key between communicating parties. One of the protocols is based on shared-key encryption and the other on public-key signatures. Both protocols rely on an on-line server rather than on certification. Needham and Schroeder also pointed out that the protocols are prone to subtle errors and that there is a need for verification techniques.

A few years later, Denning and Sacco [40] found a flaw in the shared-key Needham-Schroeder protocol and proposed their own protocol that uses a kind of name certificate issued by a trusted on-line server. More than 10 years later, both the Denning-Sacco protocol and the public-key Needham-Schroeder protocol were shown to have a serious flaws [5, 79], which well demonstrates the difficulty of designing key-establishment protocols.

For our purposes, it is interesting to note that, although the three protocols mentioned above and their failures are quite different in nature, the attacks against all three protocols are so called *replay attacks*. In all these attacks, the malicious party records legitimate protocol messages and manages to pose as someone else by replaying the recorded messages or parts of them in another context. From the taxonomy of replay attacks by Syverson [112], it is clear that the majority of protocol failures falls into this wide category.

There are three basic directions of research that have addressed the security of authentication protocols: verification tools, logical proofs of correctness, and robust engineering practices. (There are other categorizations. Detailed surveys of the area can be found in [106, 68, 84].)

3.2 Formal verification

A well-established approach to the formal verification of protocols is to model them as state machines and to apply various model-checking techniques from straightforward state-space enumeration to sophisticated proof techniques to check their properties. This line of work was started by Dolev and Yao [42] who analyzed a limited family of so called ping-pong protocols whose security can be decided in polynomial time under a very general attack model. In its most general form, the security of cryptographic protocols is naturally undecidable and the verification tools have to balance their expressive power and computability. Some early model checkers were Millen's Interrogator [90] and Meadows's NRL Protocol Analyzer [85]. Recently, the authentication protocols have become a kind of benchmark application for formal verification methods. Most types of formalisms including various process calculi, Petri-nets, Murϕ, and high-order logic have been tried [105, 4, 96, 10, 91, 99]. The interest is natural because the protocol specifications are very limited in size, often only a few messages that are exchanged in lockstep, but there are nevertheless frequent flaws to find.

These formalisms have adopted the attack model of Dolev and Yao where the protocol participants are modelled as reactive state machines that follow the protocol specification strictly in accepting and sending messages. The messages are received from and sent to a hostile environment that is in a complete control over their delivery. It can freely drop, reorder and insert messages. The environment initially possesses session keys and recorded messages from old protocol runs and private keys of some conspiring participants, and it remembers all the messages sent during the current protocol run. It may produce new messages from the old ones with rewriting rules that should accurately depict encryption, decryption and other computational capabilities of a powerful attacker.

The active research in the area has succeeded in removing many limitations of the early models. For example, automatic and interactive theorem provers and symbolic model checking techniques have been used to verify properties of protocols with infinite state spaces, unlimited number of participants and protocol runs and with arbitrarily complex messages [85, 99, 63]. Some current challenges are to extend the applicability of the models to new areas such as electronic commerce protocols and to cover new threats like traffic analysis [87].

3.3 Authentication logics

The authentication logics, pioneered by Burrow, Abadi and Needham [32], approach the problems of protocol security from a different angle. They abstract away the detailed attacker model and, instead, formalize the common ways in which cryptographic primitives are used as axioms and inference rules. The rules of the logic describe the changes in the participants' knowledge or beliefs during a protocol run. The beliefs relate directly to engineering terms such as "Now Alice knows this key is shared only by her and Bob." In developing a proof of security, one has to state the assumptions and goals of the protocol explicitly and to explain what every message achieves. Thus,

the logics can effectively be used to support protocol development. The high level of abstraction, however, means that the logic formalisms are less general and must be modified when any cryptographic algorithm is used in a new way. Soon after the original BAN logic, many variants (e.g. [56, 114]) were proposed. Recently, there has been some work towards automating the reasoning that has brought the authentication logics and formal verification tools closer to each other [29].

3.4 Protocol design principles

While formal methods help in verifying and proving the correctness of protocols, they do not answer the question of how one should design one. There is another thread of research that looks for protocol engineering principles that help in producing secure protocols. The design principles are usually derived by examining known protocol flaws and by explaining how they could have been avoided. As pointed out by Syverson [115], such principles are neither sufficient nor necessary for security. However, they help in protocol design and may significantly reduce the risk of failure.

Simmons [109] examines some protocol failures and concludes that one should enumerate all properties of all quantities involved in the protocol and assume that any combination of them may fail. In designing a protocol, one should "never assume anything that can't be enforced or verified".

The most comprehensive exposition of protocols flaws, and design principles that could prevent them, appeared in the work of Abadi and Needham [5, 1]. The protocol designer must be clear about the exact purposes of the different cryptographic functions. When the literature often ambiguously talks about encryption, one should distinguish at minimum between confidentiality protection, authentication, binding of message parts together and random number generation. *Nonces* are similarly used for many purposes and must have different properties depending on the protocol. They could be time stamps and serial numbers, or if unpredictability is desired, random numbers. The protocol designer should also examine the trust relationships between principals: what can peer entities and trusted servers be expected to do and what happens if they fail? Above all, Abadi and Needham emphasize the principle of explicit communication. The protocol flaws that they found can typically be fixed by adding the names of the protocol participants and a direction indicator to each message.

A paper of Woo and Lam [118] introduces the *full information protocols* where the "initiator and responder include in every outgoing encrypted message all the information each has gathered so far in the authentication exchange". This definition was motivated by a protocol that failed because the name of the sender was missing from one message. In order to avoid similar flaws in the future, Woo and Lam suggest explicitly including all available information in all messages. This is an extreme interpretation of the explicit communication principle. Carlsen [33] arrives at a similar solution by considering all the type information that could be added to protocol messages.

Obviously, adding redundant information in signed or otherwise protected messages can prevent many attacks as long as care is taken not to leak any secret information. The recipient of a message compares the redundant data

against what it already knows. This ensures that the principals agree on the compared data values. If the data does not match, there is a reason to suspect that the messages have been tampered with. In the Dolev and Yao model, redundant data reduces the usability of the recorded messages for the attacker: it becomes less likely that any rewriting transformation available for the malicious environment will produce a new protocol message with correct values in the redundant fields.

The main problem with the redundant information is that it consumes communication bandwidth. Designers of communication protocols prefer to optimize the message lengths, which in fact is the origin of many security flaws. Woo and Lam provide some heuristics for removing redundant information without compromising the protocol security. Nevertheless, the full information protocols remain too costly for many applications such as heavily used Internet protocols and wireless communications. Even explicit type tags (such as those in ASN.1 [64] encodings) are often considered too expensive.

Publication [**P6**] shows how redundancy can be added to protocols without causing a significant increase in message lengths. The solution is to hash the type tags and the full information from the earlier protocol messages with a one-way hash function and to include the hash value instead of the full data. The receiver can verify the hash because it already knows the redundant values. Most protocol messages already include hash values, for example, as a part of a signature, and random or pseudorandom data, such as keys, that the receiver already knows. By hashing the additional redundant data together with these fields, it can be included without increasing the message length at all. The cost of hashing is fairly low, considering that it can bind the messages and their parts to their original context and therefore is sufficient to prevent most replay attacks. The idea is based on the observation that only about 128 bits of redundancy is needed to perform a cryptographically strong check for tampering in any amount of data, and the detection of differences between the sender's and recipient's views of the protocol run does not require any more.

It is, however, necessary to exercise care when adding redundancy to the messages. As Mao and Boyd [81] point out, encrypting recognizable data with a long term secret such as a master shared key increases the danger that cryptanalysis will recover the key during its long life-time. Anderson and Needham [9, 8] come up with further design principles for protocols that apply public-key cryptography. The public-key operations often have algebraic properties that result in unexpected interaction between protocol messages and operations. Usually these can be avoided by following well-studied standard practices (e.g. [88, 73]) but sometimes new applications and combinations of algorithms may produce surprises.

While most literature on protocol design points out and suggests remedies for individual failures, Boyd and Mao [28] develop a family of secure authentication protocols systematically. Gong and Syverson [57] define fail-stop protocols, whose execution automatically halts in response to an active attack. The analysis of the protocols becomes easier because it suffices to examine passive attacks. Similarly, the full-information protocols may be easier to analyze and prove correct than arbitrary ones. Clearly, more work is needed

on the systematic design of cryptographic protocols whose correctness can be verified. Moreover, the scope of the methods should be extended to protocols with more variable goals than simple key exchange. For example, many protocols need to resist denial-of-service attacks.

4 AVAILABILITY OF SERVICES

"This Internet thing had to end sooner or later."
Matt Blaze commenting on recent network DOS attacks [24]

As the world has grown more dependent on the communications networks, even short interruptions in the functioning of the Internet and other networks have become unacceptable. Consequently, denial-of-service (DOS) attacks that prevent access to on-line services are one of the greatest threats to the information society.

Service availability may conflict with other security goals. When integrity and confidentiality are the main goals, the most secure system is often one that does nothing. For example, a server in a cryptographic protocol often responds to a suspected attack by dropping the connection and by letting the client start from the beginning. Such a response may open paths for denial-of-service attacks. Therefore, protection against DOS often requires architectural changes to the system, which may prove expensive. For the security engineer, it would be tempting to ignore attacks against service availability and think about them as a capacity planning issue. However, denial-of-service attacks, where one entity purposely denies the access to others, are clearly a security issue as they involve malicious behavior by an attacker that harms the honest users. At minimum, all security mechanisms, including cryptographic protocols, should be designed so that they do not make the system unnecessarily vulnerable to DOS.

Millen [89] divides denial-of-service attacks into two categories: resource allocation and resource destruction. We first consider attacks of the former kind, in which a malicious user intentionally exhausts resources from a shared pool with a limited capacity and, in that way, prevents others from making progress in their tasks. Publications [P7]–[P8] propose improvements to transport and authentication protocols to enhance their resistance to DOS attacks. Attacks of the latter kind are usually enabled by design or implementation errors. Usually, it should not be possible for a user to destroy a resource. However, in [P9] we consider destructive attacks that differ from this general rule. We analyze the resistance of network topologies against the breaking of connections by the destruction of communications links.

4.1 Classical DOS models

Denial-of-service by resource exhaustion can occur when several entities with conflicting interests share a resource. The first computer systems with such shared resources were multi-user operating systems where the users, or processes acting on their behalf, compete for memory, disk capacity, and processor time. Therefore, it is natural that the first models of denial-of-service concentrated on multi-user OSs.

Gligor [53] defines denial of service as a failure of the system to meet a maximum waiting time (MWT) policy. That is, the users should be granted access to a shared service within a fixed period of time from a request. It is the task of the operating system to arbitrate service requests in such a way that the MWT is met for at least some privileged group of users. The MWT policy corresponds to a fairness requirement in a real-time OS and the same

formal analysis techniques can be used to verify the fairness of both DOS-free and real-time resource allocation.

The model makes several implicit assumptions. First, there must be a reference monitor in the operating system that can control all access requests. Second, the users or processes must be assigned priorities so that the resources can be given first to privileged users. The above assumptions fit well to the multi-level secure operating systems where the priorities may be assigned in the same way as the clearances according to the user's rank in the organization. Amoroso [7] even suggests that the ability to deny a service to another user could be treated on the policy level in the same way as the ability to read and write confidential data in the Bell-LaPadula model. Another assumption these models make is, of course, that there are enough resources to satisfy all the high-priority requests from friendly users.

The DOS models described above fail when there is no TCB with a reference monitor that could control all user actions, or when there are not enough resources to satisfy even all high-priority requests within the specified maximum waiting time. Yu and Gligor [119] observed that the proofs of availability often depend on agreements between the users that are made outside the computer system and, thus, are not under the control of the TCB. The users must, for example, eventually release enough resources so that others can make progress, and reserve resources in a given order to avoid deadlocks. In order to ensure that potentially malicious users meet the agreed rules, the user requests may be filtered by trusted code, or the code may be verified at compile time. More contemporary approaches would be sandboxes and other wrappers, and signed and proof-carrying code [108, 55, 94].

One approach is to define a resource allocation policy that guarantees the availability of services to users who agree to use the resources according to some restrictive rules. Yu and Gligor, and Millen [89] define such policies. In Millen's model, a denial-of-service protection base (DPB) can revoke resources from users who do not respect certain maximum holding times. The preemptive scheduling of resources limits the kinds of tasks that users can accomplish in the system because it might sometimes be useful to hold a lock on a resource for a longer period of time. This is a common trade-off in DOS prevention: in exchange for the availability guarantees, users must accept limitations that are typically heavier than those caused by traditional security mechanisms.

4.2 DOS-resistant protocols

The classical DOS models do not extend well to open distributed systems like the Internet. The reason is again that there is no central trusted administration to set a global policy, let alone mechanisms for enforcing such policies. Furthermore, all network users are equal, with nobody to assign priorities, and there are too many simultaneous users to theoretically guarantee the availability of any service. The Dolev and Yao model of network protocol security is not applicable either. It cannot deal with DOS because it assumes that the attacker has complete control over the communications channels and hence can stop all traffic. Thus, new models are needed to understand network DOS attacks. This section first overviews some resource-exhaustion

attacks against services on open networks and possible remedies for them and then describes our new solutions in relation to the literature.

One way to consume resources from a network server is to open a large number of connections to it and to leave them all open. Many services limit the number of clients they accept simultaneously and it takes time for them to realize that a client has gone away without properly closing the connection. In the SYN attack against the TCP transport-layer protocol on the Internet [35], a malicious client sends opening messages of the protocol to the server without any intention to continue the message exchange. The server will send an acknowledgement and wait for the client's next message, which never arrives. In many TCP implementations, the half-open connections were stored in a small fixed-size table, which filled up quickly, and the failed connection attempts were purged from the table only after a relatively long timeout period. When the table was full, all connection attempts were refused. By leaving a few connections in the half-open state every minute, an attacker could effectively block all new connections from legitimate clients. What makes this attack especially attractive is that the malicious client can write a false sender address on the IP packets and, thus, mount the attack anonymously.

The resource exhausted by the SYN attack was, for implementation reasons, artificially small and easy to consume. A straightforward remedy against the attack is to increase the size of the table for half-open connections and to purge old entries from the table more readily. A more general solution to this type of attack is that the server postpones saving the protocol state until it knows that the sender address is correct. It can do this by sending the client a *cookie*, a pseudorandom number computed from the session parameters, which the client has to return in its next message. In order to return the cookie, the client has to receive the server's acknowledgement. Thus, it must be at the stated sender address or on the route to it. This and other protection mechanisms are analyzed in detail by Schuba et al. [107].

The cookie exchange at the beginning of a connection originates from the Photuris protocol by Karn and Simpson [67]. It has been adopted to other protocols including the Internet Key Exchange (IKE) by Harkins and Carrel [59]. The cookies can be seen as a weak form of authentication for the client. It gives the server enough assurance about the identity and the honest purposes of the client to justify allocating server resources, a protocol state in this case, for the client. Nevertheless, a client who is willing to reveal its location on the network and use its authentic sender address, or one who can intercept traffic from the server to read the cookies, can continue opening bogus TCP as well as application-layer connections to the server.

Since a weak authentication mitigates the problems, it might seem that strong authentication of the client could solve them once and for all. After the clients have been authenticated and their access rights checked, the server can prioritize the requests and allocate resources to the remote users according to some fair policy in the same way as in a multi-user operating system. There are several reasons why this is not a complete solution in open networks. First, there is currently no global infrastructure available for user authentication and most users do not have any credentials to prove their identities. Second, many services on the Internet are open to anyone and even ac-

tively try to attract new clients. Therefore, it may not be feasible to check the access rights before allowing access to the server. Third, the cryptographic authentication may open opportunities for new attacks. Public-key authentication protocols involve expensive exponentiation operations that can be exploited to consume server processing power. For example, the attacker may send certificates with false signatures (i.e. random bit strings in place of signatures) to the server, which has to verify every one of them. In the following, we will outline research that addresses this third problem and, hence, helps in protecting servers who do have the means for strong authentication of their clients.

Meadows [86] takes the idea of weak authentication further and suggests authenticating the client in several steps with gradually increasing cost and assurance. That way, the server will allocate more memory and computational resources when it already has some assurance of the client identity. The assurance is measured by the cost of a successful attack for the client.

[**P7**] generalizes the cookie approach in another way. We observe that stateless protocols are more reliable than stateful ones under a heavy load or under a denial-of-service attack. The paper explains how a stateless server can maintain secure connections by passing the cryptographically protected state data to the client. Admittedly, it may be unrealistic to make most protocols (and in particular, TCP, with which we experimented) completely stateless. Instead, one should consider which parts of the protocol benefit most from the added robustness. One situation where a server clearly benefits from not saving a connection state is at the beginning of an authentication protocol before the client has been reliably authenticated. A stateless authentication protocol has previously appeared in the KryptoKnight architecture by Janson et al. [65], and our paper gives more examples.

4.3 Client puzzles

Another technique that protects against misuse of the server resources comes from Dwork and Naor [44]. They suggested increasing the cost of electronic junk mail by forcing the sender of every email to solve a small puzzle before the message is accepted by the receiver. In their scheme, the receiving host generates a small cryptographic puzzle, such as the factorization of a medium-length integer, and asks the sending host to solve the puzzle. After verifying the solution, which can be done at low cost, the receiver accepts the mail. The cost of solving the puzzle is acceptable for normal users but high for mass mailers that send thousands or millions of messages.

Juels and Brainard [66] recently presented a simpler *client puzzle* that could be sent to TCP clients during a suspected SYN attack. A server that thinks it is under a denial-of-service attack can ask its clients to compute the reverse of a secure one-way function by brute force before they are allowed to open a connection. The cost of the brute force computation is parameterized by revealing some input bits to the client and by letting it find the remaining ones.

The ideas from [**P7**] and [66] were combined by Hirose and Matsuura [61] to create a DOS-resistant version of their KAP protocol. In the protocol, the client must do the first exponentiation operations. The server remains

stateless and avoids expensive computation until it has verified the result of the client's exponentiation. The verification can be done at a low cost. In this way, the client has to commit its resources first to the protocol run.

In [P8], we extend the ideas from the DOS-resistant KAP to almost any authentication protocol. While the exponentiations the client performs in KAP are a part of the particular cryptographic algorithm, our protocol uses artificial puzzles in the style of Juels and Brainard. They are also based on the partial reversal of a secure one-way hash function. We manage to reduce the cost of creating and verifying the client puzzles to the generation of one random value and to the computation of one cryptographic hash value. Our puzzles may also be broadcast periodically to the potential clients thus saving one message from the authentication in a broadcast network. The difficulty of the puzzle can be adjusted dynamically depending on the load on the server. Normally it should be set to zero, meaning that there is no puzzle to solve.

The client puzzles and statelessness protect servers that always authenticate their clients cryptographically. They prevent attacks that exploit the first messages of the authentication protocol before the client identity has been verified. For unauthenticated connections, the initial cookie exchange and client puzzles provide a limited protection. It should be noted that we have considered only attacks that exhaust the server's memory or computational power and not ones that consume communications bandwidth by flooding the server with large amounts of traffic [36]. Such attacks are currently the greatest challenge to the reliability of the Internet. Unfortunately, protection against these attacks seems to require changes not only at the servers and protocols but also in the overall network architecture.

4.4 Network topology

In addition to the protocols, the reliability of a network depends on its physical design. Reliability-engineering models are usually probabilistic and assume failures in all parts of the system to occur independently. With sufficient number of redundant components, it is possible to make the probability of a system failure arbitrarily small. Malicious attackers, on the other hand, can coordinate their attacks to cause simultaneous failures and maximal damage. Therefore, the standard reliability engineering methods cannot be directly applied to security analysis. Instead, we must consider the worst-case scenarios.

The network topology is one fairly system-independent design feature that has a heavy impact on the communications reliability. The higher the number of redundant routes between two nodes, the more work an attacker needs to do to disconnect the nodes from each other. With an adaptive routing algorithm, packets will find their way to the destination as long as there is one good route. This kind of a network can be modelled as a graph where the arcs represent network links. An attack that manages to disconnect nodes from each other corresponds to the removal of arcs in such a way that the graph is partitioned. Hence, the robustness of a network topology can be analyzed with graph-theoretical methods.

Naturally, the attacker will want to minimize the number of links it needs

to destroy or disable and to maximize the number and value of broken connections. Our goal is to find such optimal attacks against the network. In the simplest form of the problem, the attacker wants to disconnect two given nodes from each other. In that case, a MIN CUT algorithm can find the optimal set of links to destroy in a polynomial time. Cunningham [38] studied another polynomial-time version of the problem where the objective is to divide the network into several partitions and to minimize the cost per created partition. Cunningham went on to investigate optimal improvements to the network topology to make the partitioning as expensive as possible.

Several other variations of the network partitioning problem are NP complete. Dahlhaus et al. [39] show this for the *multi-way cut* where three or more given nodes should be disconnected from each other. Phillips defines the *network inhibition* [100] version of MIN CUT, where links may be partially disabled and their capacity decreases linearly with the cost to the attacker. In this model, the problem of minimizing the residual network capacity between two nodes with a given attack budget is also NP complete.

In [**P9**], we consider a new kind of attack where the goal is to disconnect a single *source node* from as many other network nodes as possible by removing a certain number of links. This is a common scenario when the robustness of a network topology is evaluated from the perspective of one server or one client. We assume adaptive routing, infinite link capacity, and that links either fail or function completely. Hence, the problem is to find a cut of the network where the partition containing the source node has a minimal size. The links may be given weights according to how expensive it is for the attacker to disable them. The nodes may be assigned values so that the damage caused by an attack is calculated as the total value of the nodes disconnected from the source. We show this problem to be NP complete even when all the weights equal 1.

Nevertheless, we manage to find optimal attacks in practice by encoding the network as a logic program with stable-model semantics [52] and by finding stable models, which correspond to attacks, with a general-purpose model finder *smodels* designed by Niemelä and Simons [97]. These results can be used in the design of robust network topologies. We found the logic-programming tool particularly useful for early prototype models where the exact definition of the system is still uncertain. While small changes in the problem definition may require great modifications to graph-theoretical algorithms, only small adjustment to the logic-program model is needed. Now that the problem is well defined, there is a potential for graph algorithms that solve it directly. Furthermore, algorithms for finding approximate solutions should be investigated, as they may give a reasonably accurate idea of the network robustness.

Publication [**P9**] also summarizes the principles that we found helpful for analyzing denial-of-service attacks in open networks. In short, one should estimate the cost of an optimally planned attack for the desired amount of damage, or the maximal amount of damage for a given attack budget. A full picture of the robustness of a system against some type of DOS attack is obtained by plotting the damage caused by optimal attacks against their cost. For example, the number of failed links represents the cost of an attack and the number of nodes disconnected from the source node is a measure

of the damage. Similar ideas appeared in [86] and in [**P8**] where a protocol is considered robust against DOS if the cost of an attack (i.e. resources used by a malicious client) is always at least as high as, or in some other desired proportion to, the damage (i.e. resources consumed from the server).

5 CONCLUSION

The publications included in this dissertation discuss several security issues and mechanisms which are new or have been highlighted on the Internet and other open communications networks where mutually distrusting entities must share resources and co-operate. The emphasis is on decentralized, locally implementable techniques that do not require global authorities or support infrastructure.

We defined a formal model of key-oriented access control and used this model to develop algorithms for access-control decisions from a certificate database. We also surveyed privacy protection in public-key infrastructures, introduced a new kind of threshold certificate, and presented novel certificate-based solutions for access control between mutually distrusting software packages on intelligent-network routers and for software license management with smartcards.

The key-oriented access control is incremental work that builds on the long tradition of computer security research. Some theoretical problems remain, such as the optimal balance between the expressiveness of certificates and the computational cost of their use, but the the main challenges for future research in this direction lie in the applications.

We also described novel design principles for cryptographic protocols to improve their robustness against common replay attacks at a low cost and to protect on-line services against denial-of-service attacks that attempt to exhaust server memory and computational resources. Stateless protocols where the clients commit their resources before the server were found to be robust against many resource exhaustion attacks. Additionally, we developed a method for analyzing the vulnerability of network topologies to denial of service by the destruction of communications links.

Denial-of-service attacks are among of the greatest threats facing the global information infrastructure. There are still many open and yet undiscovered problems in this area. General design principles for improving the availability of services on open networks are in great demand.

A CORRECTIONS AND ADDITIONS TO THE PUBLICATIONS

On the structure of delegation networks [**P2**]:

- Although not defined with key-oriented certificates in mind, the calculus of Abadi et al. [3, 74, 117] is expressive enough to be used for reasoning about chains of key-oriented certificates. (Sec. 1 of [**P2**] is incorrect on this point.)

  The certificates in a chain can be encoded as Taos *general delegation certificates*. The access control list must have an entry of the form $K_n \ for \ldots for \ K_2 \ for \ K_1$ that matches all the principals in the delegation chain. However, the Taos implementation supports an iteration construct in the ACL that can match an arbitrary chain ([117], Sec. 7.3).

  In Taos, *roles* are used to restrict the delegated access rights. It is rather impractical to describe the authorization fields of SPKI-like certificates with roles. Howell [62] has recently extended the calculus of Abadi et al. so that restricted delegation can be directly reasoned about.

Fast access control decisions from delegation certificate databases [**P3**]:

- The two-way search could also be used in implementing a KeyNote policy compliance checker. ([**P2**] and [**P3**] predate the KeyNote publications.) A KeyNote query includes a set of credentials in arbitrary order and it is the task of the compliance checker to construct a directed graph of them to show that an action conforms to the policy, or to decide that no such graph exists. A KeyNote implementation may also have access to an external certificate database, for example, through call-back functions. The KeyNote specification [25] leaves the algorithms open as an implementation issue.

Privacy and accountability in certificate systems [**P4**]:

- Sec. 3.4 of [**P4**] says "...the increased number of certificates does not require additional storage or communications capacity when they are distributed to the trustees." This is true if a copy of the normal threshold certificate would be distributed to and stored by every trustee, which may not be necessary in all applications. On the other hand, the improved privacy of our open threshold certificates depends on the subcertificates being stored by the individual trustees.

  It should also be noted that additional communications capacity is needed at the time when the subcertificates are used. This is because the co-operating trustees must send their subcertificates to the verifier. However, this does not require additional messages if the trustees attach their subcertificates to the access request or to the delegation certificates, which they must sign and pass to the server in any case.

Software license management with smart cards [**P5**]:

- If the tamper-proof modules (i.e. smartcards) are capable of generating new signature keys, old, empty cards may be reused and need not be discarded. (This was pointed out by Michael Roe.) Key-generation requires a tamper-proof randomness source on the card, which may be expensive to implement. The value of this feature in our system would be limited since license transfer is allowed only from older cards to newer ones.

Stateless connections [**P7**]:

- We have implemented a transport-layer protocol where the server (i.e. the sender) is completely stateless. The client (i.e. the receiver) maintains a sliding window and practices TCP-compatible congestion avoidance. The performance of the protocol is comparable to TCP. The server is extremely simple while complexity is increased on the client side.

DOS-resistant authentication with client puzzles [**P8**]:

- Before [66], the reversal of one-way functions with k unknown bits in the argument has appeared in various publications.

  Manber [80] suggests an improvement to the protection of Unix-style public passwords files that contain the hash value of each password with a random *salt*. In Manber's scheme, the host keeps some bits of the salt secret, erases them from its memory, and searches for them by brute force every time the password is verified. An attacker trying to crack passwords from the file will experience the same proportional increase in the cost of computation.

  Syverson [113] uses similar puzzles (called *weakly secret bit commitment*) to keep the result of a lottery secret for a predictable amount of time. Rivest et al. [104] present a puzzle based on public-key cryptography that is believed to require an inherently serial computation to solve. Hence, they can control more accurately the time it takes to solve the puzzle. In our application, we want to control the cost of solving the puzzle and not the time. Therefore, we prefer the simpler puzzles based on secure hash functions, which may be solved quickly by distributing the computation to a large number of processors.

- Unlike our puzzles, the client puzzles of Juels and Brainard [66] may consist of several subpuzzles, which must all be solved by the client. In addition to the finer granularity of puzzle complexity, the advantage of subpuzzles is that the cost of solving a single puzzle can be predicted more accurately. (If a puzzle is divided into subpuzzles so that the expected number of trials needed to solve the puzzle remains unchanged, the variance of the number of trials becomes smaller.) In a flooding attack, however, the attacker will need to solve a large number of puzzles to cause any significant damage, and the total cost can be predicted fairly accurately no matter how the cost of a single puzzle is distributed. Therefore, we are satisfied with the simpler puzzle.

Analyzing single-server network inhibition [**P9**]:

- Definition 2, page 110: $A_N \times L \cup L \times A_N$ should be $A_N \times \mathbf{N} \cup \mathbf{N} \times A_N$.
  Problem 6, page 111: $N \cap L$ should be $N \cup L$.

References

[1] Martín Abadi. Explicit communication revisited: two new attacks on authentication protocols. *IEEE Transactionson Software Engineering*, 23(3):185–186, March 1997.

[2] Martín Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1–2):3–21, October 1998.

[3] Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, September 1993.

[4] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: the spi calculus. *Information and Computation*, 143(1):1–70, January 1999.

[5] Martín Abadi and Roger M. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, January 1996. Appeared first in Proc. 1994 IEEE CS Symposium on Research in Security and Privacy.

[6] Marshall D. Abrams. Renewed understanding of access control policies. In *Proc. 16th National Computer Security Conference*, pages 87–95, Baltimore, MD USA, September 1993. NIST / NCSC.

[7] Edward Amoroso. A policy model for denial of service. In *Proc. Computer Security Foundations Workshop III*, pages 110–114, Franconia, NH USA, June 1990. IEEE Computer Society Press.

[8] Ross Anderson and Roger Needham. *Programming Satan's Computer*, volume 1000 of *LNCS*, pages 426–440. Springer, 1995.

[9] Ross Anderson and Roger Needham. Robustness principles for public key protocols. In *Advances in Cryptology - Proc. CRYPTO '95*, volume 963, pages 236–247, Santa Barbara, CA USA, August 1995. Springer.

[10] Tuomas Aura. Modelling the Needham-Schröder authentication protocol with high level Petri nets. Technical Report B14, Helsinki University of Technology, Digital Systems Labratory, Espoo, Finland, September 1995.

[11] Tuomas Aura. On the structure of delegation networks, Licentiate's thesis. Technical Report A48, Helsinki University of Technology, Digital Systems laboratory, Espoo, Finland, December 1997.

[12] Tuomas Aura. Strategies against replay attacks. In *Proc. 10th IEEE Computer Security Foundations Workshop*, pages 59–68, Rockport, MA USA, June 1997. IEEE Computer Society Press.

[13] Tuomas Aura. Fast access control decisions from delegation certificate databases. In *Proc. 3rd Australasian Conference on Information Security and Privacy (ACISP '98)*, volume 1438 of *LNCS*, pages 284–295, Brisbane, Australia, July 1998. Springer.

[14] Tuomas Aura. On the structure of delegation networks. In *Proc. 11th IEEE Computer Security Foundations Workshop*, pages 14–26, Rockport, MA USA, June 1998. IEEE Computer Society Press.

[15] Tuomas Aura. Distributed access-rights management with delegation certificates. In *Secure Internet Programming – Security Issues for Distributed and Mobile Objects*, volume 1603 of *LNCS*, pages 211–235. Springer, 1999.

[16] Tuomas Aura, Matt Bishop, and Dean Sniegowski. Analyzing single-server network inhibition. In *Proc. 13th IEEE Computer Security Foundations Workshop*, pages 108–117, Cambridge, UK, June 2000. IEEE Computer Society Press.

[17] Tuomas Aura and Carl Ellison. Privacy and accountability in certificate systems. Reseach Report A61, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, April 2000.

[18] Tuomas Aura and Dieter Gollmann. Software license management with smart cards. In *Proc. USENIX Workshop on Smartcard Technology*, pages 75–85, Chicago, IL USA, May 1999. USENIX Association.

[19] Tuomas Aura and Pekka Nikander. Stateless connections. In *Proc. International Conference on Information and Communications Security (ICICS'97)*, volume 1334 of *LNCS*, pages 87–97, Beijing, China, November 1997. Springer Verlag.

[20] Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo. DOS-resistant authentication with client puzzles. In *Proc. Security Protocols Workshop 2000*, Cambridge, UK, 2000. Springer. To appear.

[21] D. Elliott Bell and Leonard. J. LaPadula. Secure computer systems: unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, The Mitre Corporation, Bedford, MA USA, March 1976.

[22] Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *Computer Security - Proc. ESORICS 94*, volume 875 of *LNCS*, pages 3–17, Brighton, UK, November 1994. Springer.

[23] K. Biba. Integrity considerations for secure computer systems. Technical Report MTR-3153, The Mitre Corp., Bedford, MA USA, 1975.

[24] Matt Blaze. Computer Security Protocols workshop, panel on network denial of service, Cambridge, UK, April 2000.

[25] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos Keromytis. The KeyNote trust-management system version 2. RFC 2704, IETF Network Working Group, September 1999.

[26] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, volume 1603 of *LNCS*, pages 185–210. Springer, 1999.

[27] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proc. 1996 IEEE Symposium on Security and Privacy*, pages 164–173, Oakland, CA USA, May 1996. IEEE Computer Society Press.

[28] Colin Boyd and Wenbo Mao. Designing secure key exhange protocols. In *Computer Security - Proc. ESORICS 94*, volume 875 of *LNCS*, pages 93–105, Brighton, UK, November 1994. Springer.

[29] Stephen Brackin. A HOL extension of GNY for automatically analyzing cryptographic protocols. In *Proc. 9th IEEE Computer Society Computer Security Foundations Workshop*, pages 62–77, Kenmare, Ireland, June 1996. IEEE Computer Society Press.

[30] David F. C. Brewer and Michael J. Nash. The Chinese wall security policy. In *Proc. 1989 IEEE Symposium on Research in Security and Privacy*, pages 206–214, Oakland, CA USA, May 1989. IEEE Computer Society Press.

[31] John A. Bull, Li Gong, and Karen R. Sollins. Towards security in an open systems federation. In *Computer Security - Proc. ESORICS 92*, volume 648 of *LNCS*, pages 3–20, Toulouse, France, November 1992. Springer-Verlag.

[32] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.

[33] Ulf Carlsen. Cryptographic protocol flaws. In *Proc. IEEE Computer Security Foundations Workshop VII*, pages 192–200, Franconia, NH USA, June 1994. IEEE Computer Society Press.

[34] CCITT. *Recommendation X.509, The Directory - Authentication Framework*, volume VIII of *CCITT Blue Book*, pages 48–81. 1988.

[35] CERT. TCP SYN flooding and IP spoofing attack. CERT Advisory CA-96.21, November 1996.

[36] *Results of the Distributed-Systems Intruder Tools Workshop*, Pittsburgh, PA USA, December 1999. The CERT Coordination Center.

[37] D. Clark and D. Wilson. A comparison of commercial and military computer security policies. In *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA USA, 1987. IEEE Computer Society Press.

[38] William H. Cunningham. Optimal attack and reinforcement of a network. *Journal of the ACM*, 32(3):549–561, July 1985.

[39] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts (extended abstract). In *Proc. 24th Annual ACM Symposium on Theory of Computing (STOC'92)*, pages 241–251, Victoria, Canada, May 1992. ACM Press.

[40] Dorothy E. Denning and Giovanni Maria Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533–536, August 1981.

[41] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[42] D. Dolev and A. Yao. On the security of public-key protocols. *Communications of the ACM*, 29(8):198–208, August 1983.

[43] Deborah. D. Downs, Jerzy. R. Rub, Kenneth. C. Kung, and Carole. S. Jordan. Issues in discretionary access control. In *Proc. 1985 Symposium on Security and privacy (SSP '85)*, pages 208–218, Oakland, CA USA, April 1985. IEEE Computer Society Press.

[44] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology - Proc. CRYPTO '98*, volume 740 of *LNCS*, pages 139–147, Santa Barbara, CA USA, August 1992. Springer-Verlag.

[45] Jean-Emile Elien. Certificate discovery using SPKI/SDSI 2.0 certificates. Master's thesis, Massachusetts Institute of Technology, May 1998.

[46] Carl Ellison. Generalized certificates. Unpublished notes, March 1996.

[47] Carl Ellison. SPKI requirements. RFC 2692, IETF Network Working Group, September 1999.

[48] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. SPKI examples. Internet draft, March 1998.

[49] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. Simple public key certificate. Internet draft, July 1999.

[50] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. SPKI certificate theory. RFC 2693, IETF Network Working Group, September 1999.

[51] Carl Ellison and Bruce Schneier. Ten risks of PKI: what you are not being told about public key infrastructure. *Computer Security Journal*, XVI(1), 2000.

[52] M. Gelfold and V. Lifschitz. The stable model semantics for logic programming. In *Proc. 5th International Conference on Logic Programming*, pages 1070–1080, Seattle, WA USA, August 1988. The MIT Press.

[53] Virgil D. Gligor. A note on the denial-of-service problem. In *Proc. 1983 IEEE Symposium on Research in Security and Privacy*, pages 139–149, Oakland, CA USA, April 1983. IEEE Computer Society.

[54] Li Gong. A secure identity-based capability system. In *Proc. 1989 IEEE Symposium on Research in Security and Privacy*, pages 56–63, Oakland, CA USA, May 1989. IEEE Computer Society Press.

[55] Li Gong. *Inside Java 2 Platform Security*. Addison-Wesley, 1999.

[56] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *Proc. 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 234–248, Oakland, CA USA, May 1990. IEEE Computer Society Press.

[57] Li Gong and Paul F. Syverson. Fail-stop protocols: an approach to designing secure protocols. In *Proc. 5th International Working Conference on Dependable Computing for Critical Applications (DCCA-5)*, pages 44–55, Urbana, IL USA, September 1995. Springer.

[58] Joseph Y. Halpern and Ron van der Meyden. A logic for SDSI's linked local name spaces (preliminary version). In *12th IEEE Computer Security Foundations Workshop*, pages 111–122, Mordano, Italy, June 1999. IEEE Computer Society Press.

[59] Dan Harkins and Dave Carrel. The Internet key exchange (IKE). RFC 2409, IETF Network Working Group, November 1998.

[60] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, August 1976.

[61] Shouichi Hirose and Kanta Matsuura. Enhancing the resistance of a provably secure key agreement protocol to a denial-of-service attack. In *Proc. 2nd International Conference on Information and Communication Security (ICICS'99)*, pages 169–182, Sydney, Australia, November 1999. Springer.

[62] Jonathan R. Howell. *Naming and sharing resources across administrative boundaries*. PhD thesis, Darthmouth college, May 2000.

[63] Antti Huima. Analysis of cryptographic protocols via symbolic state space enumeration. Master's thesis, Helsinki University of Technology, 1999.

[64] ITU-T. Abstract syntax notation one (ASN.1). Standard, ITU-T Rec. X.680 (1997) and ISO/IEC 8824-1:1998, 1997.

[65] Philippe Janson, Gene Tsudik, and Moti Yung. Scalability and flexibility in authentication services: the KryptoKnight approach. In *IEEE INFOCOM'97*, Tokyo, April 1997.

[66] Ari Juels and John Brainard. Client puzzles: a cryptographic countermeasure against connection depletion attacks. In *Proc. 1999 Network and Distributed Systems Security Symposium (NDSS)*, pages 151–165, San Diego, CA USA, February 1999. Internet Society.

[67] Phil Karn and William A. Simpson. Photuris: session-key management protocol. RFC 2522, IETF Network Working Group, March 1999.

[68] Richard Kemmerer, Catherine Meadows, and Jonathan Millen. Three systems for cryptographic protocol analysis. *Journal of Cryptology*, 7:79–130, 1994.

[69] Joe Kilian and Erez Petrank. Identity escrow. In *Advances in Cryptology - Proc. CRYPTO '98*, volume 1462 of *LNCS*, pages 169–185, Santa Barbara, CA USA, August 1998. Springer.

[70] Reto Kohlas and Ueli Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *Proc. International Workshop on Theory and Practice of Public-Key Cryptography (PKC'00)*, volume 1751 of *LNCS*, pages 93–112, Melbourne, victoria, Australia, January 2000. Springer.

[71] Reto Kohlas and Ueli Maurer. Reasoning about public-key certification: on bindings between entities and public keys. *IEEE Journal on Selected Areas in Communications*, 18(4), April 2000. Also in Proc. Fincancial Cryptography (FC99), LNCS 1648, pp. 86–103, Springer 1999.

[72] Loren M. Kohnfelder. Towards a practical public-key cryptosystem. Bachelor's thesis, Massachusetts Institute of Technology, Cambridge, MA USA, 1978.

[73] RSA Laboratories. Public-key cryptography standards (PKCS#1–#15), 2000.

[74] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobbler. Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, November 1992.

[75] Butler W. Lampson. Protection. In *Proc. 5th Princeton Symposium on Information Sciences and Systems*, pages 437–443, Princeton, NJ USA, March 1971. Reprinted in Operating Systems Review 8(1), January 1974, pp. 18–24.

[76] Butler W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, October 1973.

[77] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. A logic-based knowledge representation for authorization with delegation. Technical Report RC 21492, IBM Research Division, May 1999.

[78] Steven B. Lipner. Non-discretionary controls for commercial applications. In *Proc. 1982 Symposium on Security and Privacy*, pages 2–10, Oakland, CA USA, April 1982. IEEE Computer Society Press.

[79] Gavin Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1996.

[80] Udi Manber. A simple scheme to make passwords based on one-way functions much harder to crack. *Computers & Security*, 15(2):171–176, 1996.

[81] Wenbo Mao and Colin A. Boyd. Development of authentication protocols: Some misconceptions and a new approach. In *Proc. IEEE Computer Security Foundations Workshop VII*, pages 178–186, Franconia, NH USA, June 1994. IEEE Computer Society Press.

[82] Ueli M. Maurer. Modelling a public-key infrastructure. In *Computer Security - Proc. ESORICS 96*, volume 1146 of *LNCS*, pages 325–350, Rome, Italy, September 1996. Springer.

[83] Catherine Jensen McCollum, Judith R. Messing, and Lou Anna Notargiacomo. Beyond the pale of MAC and DAC: Defining new forms of access control. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 190–200, Oakland, CA USA, May 1990. IEEE Computer Society Press.

[84] Catherine Meadows. Formal verification of cryptographic protocols: a survey. In *Advances in Cryptology - Asiacrypt '94*, volume 917 of *LNCS*, pages 133–150. Springer 1995, December 1994.

[85] Catherine Meadows. The NRL protocol analyzer: an overview. *Journal of Logic Programming*, 26(2):113–131, February 1996.

[86] Catherine Meadows. A formal framework and evaluation method for network denial of service. In *Proc. 12th IEEE Computer Security Foundations Workshop*, pages 4–13, Mordano, Italy, June 1999. IEEE Computer Society.

[87] Catherine Meadows. Open issues in formal methods for cryptographic protocol analysis. In *Proc. DARPA Information Survivability Conference and Exposition (DISCEX) 2000*, pages 237–250, Hilton Head Island, SC USA, January 2000. IEEE Computer Society Press.

[88] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[89] Jonathan K. Millen. A resource allocation model for denial of service. In *Proc. 1992 IEEE Computer Society Symposium on Security and Privacy*, pages 137–147, Oakland, CA USA, May 1992. IEEE Computer Society Press.

[90] Jonathan K. Millen, Sidney C. Clark, and Sheryl B. Freedman. The Interrogator: protocol security analysis. *IEEE Transactions on Software Engineering*, 13(2):274–288, February 1987.

[91] John C. Mitchell, Mark Mitchell, and Ulrich Stern. Automated analysis of cryptographic protocols using Murϕ. In *Proc. 1997 IEEE Symposium on Security and Privacy*, pages 141–151, Oakland, CA USA, May 1997. IEEE Computer Society Press.

[92] NCSC. DoD trusted computer system evaluation criteria. Report 5200.28-STD, National Computer Security Center, December 1985. (Rainbow series, Orange book).

[93] NCSC. A guide to understanding discretionary access control in trusted systems. Report NCSC-TG-003 version-1, National Computer Security Center, September 1987. (Rainbow series, Neon orange book).

[94] George C. Necula and Peter Lee. Safe, untrusted agents using proof-carrying code. In *Mobile Agents and Security*, volume 1419 of *LNCS*, pages 61–91. Springer, 1998.

[95] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, December 1978.

[96] Benjamin B. Nieh and Stafford E. Tavares. Modelling and analyzing cryptographic protocols using Petri nets. In *Advances in Cryptology–AUSCRYPT'92, International Conference on Cryptology*, pages 275–295. LNCS, Spring-Verlag, 1992.

[97] Ilkka Niemelä and Patrik Simons. Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In *Proc. 4th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 1265 of *LNCS*, pages 420–429, Dagstuhl, Germany, July 1997. Springer.

[98] Pekka Nikander and Lea Viljanen. Storing and retrieving Internet certificates. In *Proc. 3rd Nordic Workshop on Secure IT Systems (NORDSEC'98)*, Trondheim, Norway, November 1998.

[99] Lawrence C. Paulson. Proving properties of security protocols by induction. In *Proc. 10th Computer Security Foundations Workshop*, pages 70–83, Rockport, MA CA, June 1997. IEEE Computer Society Press.

[100] Cynthia A. Phillips. The network inhibition problem. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 776–785. ACM Press, May 1993.

[101] Michael K. Reiter and Stuart G. Stubblebine. Path independence for authentication in large-scale systems. In *Proc. 4th ACM Conference on Computer and Communications Security*, pages 57–66, Zürich, Switzerland, April 1997. ACM Press.

[102] Michael K. Reiter and Stuart G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2):138–158, May 1999.

[103] Ronald L. Rivest and Butler Lampson. SDSI - a simple distributed security infrastucture. Technical report, April 1996.

[104] Ronald L. Rivest, Adi Shamir, and David A. Wagner. Time-lock puzzles and timed-release crypto. Unpublished., March 1996.

[105] A. W. Roscoe. Modelling and verifying key-exchange protocols using CSP and FDR. In *Proc. 8th Computer Security Foundations Workshop*, pages 98–107, Kenmare, Ireland, June 1995. IEEE Computer Society Press.

[106] Avi D. Rubin and Peter Honeyman. Formal methods for the analysis of authentication protocols. CITI Technical Report 93-7, Center for Information Techology Integration, University of Michigan, Ann Arbor, MI USA, November 1993.

[107] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spaffold, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on TCP. In *Proc. 1997 IEEE Symposium on Security and Privacy*, pages 208–223, Oakland, CA USA, May 1997. IEEE Computer Society Press.

[108] Peter Sewell and Jan Vitek. Secure composition of insecure components. In *Proc. 12th IEEE Computer Security Foundations Workshop*, pages 136–150, Mordano, Italy, June 1999. IEEE Computer Society Press.

[109] Gustavus J. Simmons. Cryptanalysis and protocol failures. *Communications of the ACM*, 37(11):56–65, November 1994.

[110] Lawrence Snyder. Formal models of capability-based protection systems. *IEEE Transactions on Computers*, C-30(3):172–181, March 1981.

[111] Karen R. Sollins. Cascaded authentication. In *Proc. 1988 IEEE Symposium on Security and Privacy*, pages 156–163, Oakland, CA USA, April 1988. IEEE Computer Society Press.

[112] Paul Syverson. A taxonomy of replay attacks. In *Proc. IEEE Computer Security Foundations Workshop VII*, pages 131–136. IEEE Computer Society Press, June 1994.

[113] Paul Syverson. Weakly secret bit commitment: applications to lotteries and fair exchange. In *Proc. 11th IEEE Computer Security Foundations Workshop (CSFW 11)*, pages 2–13, Rockport, MA USA, June 1998. IEEE Computer Society.

[114] Paul Syverson and Paul C. van Oorschot. On unifying some cryptographic protocol logics. In *Proc. 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 14–28, Oakland, CA USA, May 1994. IEEE Computer Society Press.

[115] Paul F. Syverson. Limitations on design principles for public key protocols. In *Proc. 1996 IEEE Symposium on Security and Privacy*, pages 62–73, Oakland, CA USA, May 1996. IEEE Computer Society Press.

[116] Tom Weckström. AAA architecture for hierarchical wireless mobile IPv4. Master's thesis, Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Espoo, Finland, July 2000.

[117] Edward P. Wobber, Martín Abadi, Michael Burrows, and Butler Lampson. Authentication in the Taos operating system. *ACM Transactions on Computer Systems*, 12(1):3–32, February 1994.

[118] Thomas Y. C. Woo and Simon S. Lam. A lesson on authentication protocol design. *Operating Systems Review*, 28(3):24–37, July 1994.

[119] Che-Fn Yu and Virgil D. Gligor. A formal specification and verification method for the prevention of denial of service. In *Proc. 1988 IEEE Symposium on Security and Privacy*, pages 187–202, Oakland, CA USA, April 1988. IEEE Computer Society Press.

[120] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press, June 1995.

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A51 Kimmo Varpaaniemi
On the Stubborn Set Method in Reduced State Space Generation. May 1998.

HUT-TCS-A52 Ilkka Niemelä, Torsten Schaub (Eds.)
Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning. May 1998.

HUT-TCS-A53 Stefan Rönn
Semantics of Semaphores. 1998.

HUT-TCS-A54 Antti Huima
Analysis of Cryptographic Protocols via Symbolic State Space Enumeration. August 1999.

HUT-TCS-A55 Tommi Syrjänen
A Rule-Based Formal Model For Software Configuration. December 1999.

HUT-TCS-A56 Keijo Heljanko
Deadlock and Reachability Checking with Finite Complete Prefixes. December 1999.

HUT-TCS-A57 Tommi Junttila
Detecting and Exploiting Data Type Symmetries of .... December 1999.

HUT-TCS-A58 Patrik Simons
Extending and Implementing the Stable Model Semantics. April 2000.

HUT-TCS-A59 Tommi Junttila
Computational Complexity of the Place/Transition-Net Symmetry Reduction Method. April 2000.

HUT-TCS-A60 Javier Esparza, Keijo Heljanko
A New Unfolding Approach to LTL Model Checking. April 2000.

HUT-TCS-A61 Tuomas Aura, Carl Ellison
Privacy and accountability in certificate systems. April 2000.

HUT-TCS-A62 Kari J. Nurmela, Patric R. J. Östergård
Covering a Square with up to 30 Equal Circles. June 2000.

HUT-TCS-A63 Nisse Husberg, Tomi Janhunen, Ilkka Niemelä (Eds.)
Leksa Notes in Computer Science. October 2000.

HUT-TCS-A64 Tuomas Aura
Authorization and availability - aspects of open network security. November 2000.

ISBN 951-22-5218-X
ISSN 1457-7615