Security in TINA

S. Staamann [1] *L. Buttyán* [2] *U. Wilhelm* [1]
Swiss Federal Institute of Technology - Lausanne (EPFL)
[1] *Operating Systems Laboratory (LSE)*
[2] *Institute for computer Communications and Applications (ICA)*
EPFL-DI-{LSE, ICA}, CH-1015 Lausanne, Switzerland,
fon: +41 21 693 5267, fax: +41 21 693 6770,
email: {Sebastian.Staamann, Levente.Buttyan, Uwe.Wilhelm}@epfl.ch

Abstract

TINA is a specification of an open architecture for telecommunication services in the broadband, multimedia, and information era. Its characteristics most relevant for security are a variety of services, a multitude of service providers, a well defined business model, a middleware platform for service development and provision, and the assumption of advanced costumer premises equipment. Concepts for its security architecture are developed in the CrySTINA project. We introduce the TINA-C architecture, analyse it with regard to security, and present the CrySTINA security architecture. CrySTINA is aligned with the OMG's CORBA Security specification, but enhances it with regard to security interoperability despite the heterogeneity of security policies and technologies that must be expected in TINA networks. Thus, we present a model for the enforcement of security policies that supports the negotiation of security contexts.

Keywords

TINA, Security, CORBA, DPE, Interoperability, Security Context

1 INTRODUCTION

Computers are increasingly used for the control and operation of telecommunication systems. The use of computers enables more flexibility in control as well as the fast and cheap introduction of new telecommunication services. The basic ideas for this approach stem from the Intelligent Network (Garrahan *et al.* 1993) (Magedanz and Popescu-Zeletin 1996). Network and service control functions of these networks are more and more realized as software. TINA, which stands for Telecommunications Information Networking Architecture (Dupuy *et al.* 1995), takes these developments even further. It is currently the most encompassing effort to define an open architecture for telecommunication services implemented as distributed applications in the emerging broadband, multimedia and information era. This effort is carried out by the TINA Consortium (TINA-C), a multinational consortium consisting of major network operators, as well as telecommunication equipment and computer system suppliers (Barr *et al.* 1993). The software model of TINA is based on the concept of distributed object computing. The middleware in TINA will be based on products conforming to the Common Object Request Broker Architecture (CORBA) specified by the OMG (OMG 1995_1).

In the CrySTINA project, we develop a security architecture for TINA that is aligned with the CORBA security architecture, but additionally supports the negotiation of security contexts. In this article, we present this architecture and the rationale behind it. Section 2 introduces the TINA-C architecture focusing on the security relevant aspects. In Section 3, we analyse TINA with regard to security, and introduce the CORBA security specifications. In Section 4, we present the CrySTINA security architecture and describe a model for the enforcement of security. Section 5 summarizes the work presented and gives an outlook on ongoing and further work. Familiarity with TINA is not required, while knowledge about CORBA and security concepts is assumed. The full text, which is presented here in an abridged version, is available as a Technical Report (Buttyán *et al.* 1998).

2 THE TINA-C ARCHITECTURE

In TINA, services are realized as distributed applications (Figure 1). They consist of service components that interact with each other via a Distributed Processing Environment (DPE). The DPE is a software layer that operates above the Native Computing and Communications Environment (NCCE), which is an abstraction of the computing hardware and the operating system of the service nodes. While the NCCE is technology dependent, the DPE offers a uniform interface to the distributed environment. The DPE will consist of CORBA implementations and additional TINA specific services.

Service components are specified in TINA-C's Object Description Language (ODL), which is a superset of CORBA's Interface Description Language (IDL). They consist of computational objects (COs) or CO groups. COs can have two kinds of interfaces: operational interfaces, which are comparable to object interfaces in CORBA, and stream interfaces, which are designed to convey an arbitrary sequence of bytes between two components (e.g., audio or video bit streams). Messages from and to the operational interfaces are exchanged via the Kernel Transport Network (KTN), whereas streams are transferred via the Transport Network of the telecommunication operator.

TINA has a general business model with various roles for the stakeholders: the supermarket model. Nevertheless, any arbitrarily complex relationship between stakeholders is composed of simple two-party user-provider relationships. A user-provider relationship contains two types of interaction: access and usage. The access part is concerned with the establishment of a trusted

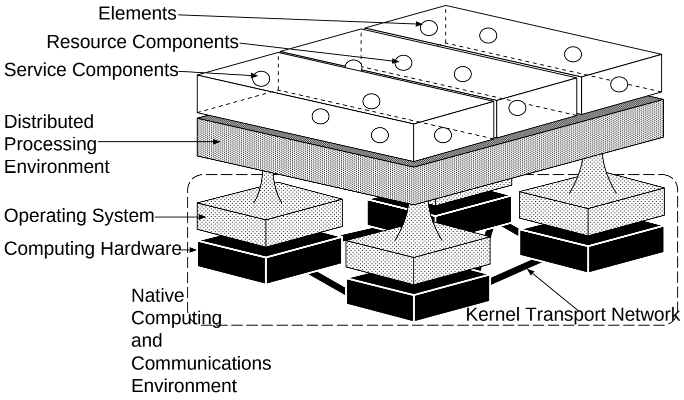

Figure 1 TINA overall architecture

and reliable temporary relationship between the user domain and the provider domain; this is a prerequisite for the usage part, which is concerned with the provision of the actual service.

The traditional call concept of telecommunications is substituted by the more flexible session concept. At the service control level, there are access sessions and service sessions (the latter concerned with the service usage). At the communication control level, a communication session controlled by the service session is responsible for providing the necessary stream connections for contents delivery. The service session represents the instances of service usage and the state of the service logic. Exactly one provider is involved in a service session, but one or more users participate in it. Before being able to participate in a service session, each user must establish an access session with the provider. This is comparable with a login session on a multiuser computer.

Sessions and other information objects are mapped onto service components. Service control is achieved by the interaction of service components in the administrative domains of the users and the provider. Figure 2 shows the service components in their administrative domains and their relation to different sessions using the example of a video conference with two participants. For a full description of all service components, please refer to the TINA Service Architecture document (TINA 1997).

Most relevant for security are the service components for the access session, namely the Provider Agent (PA) in the user domain, as well as the Initial Agent (IA) and the User Agent (UA) in the provider domain, since authentication takes place between these components. The UA represents the user in the provider domain. It is the user's contact point to start or resume a service session. In order to interact with his UA (i.e., to have an access session), the user contacts the IA using the PA. The IA is the initial contact point of the provider for all users. Authentication between the user and the provider

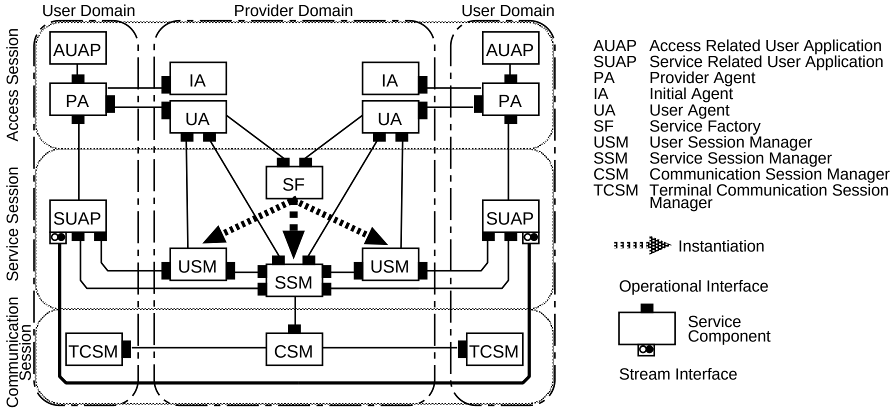

Figure 2 TINA service architecture

is then performed between the PA and the IA as part of the establishment of the access session. After successful authentication, a reference to the UA is delivered to the PA. Additionally, a security association for further interactions between service session related service components should have been established.

3 SECURITY ANALYSIS

Structuring the Security Problem Domain:
Security concerns all parts of a TINA system; it is pervasive and cannot be addressed in isolation. To cope with this complexity, it is necessary to structure the security problem domain in an appropriate way. We structure the security problem domain according to two criteria: the architectural levels defined in the overall architecture, and the type of information (the network used for the transport), i.e., control messages (KTN) or communication contents (Transport Network). We identified the following subdomains of the security problem domain:
System Security shall ensure that systems, mainly the hardware and the operating system, are not subject to intrusions or modifications. This concerns networking resources (e.g., network switches) and computing resources. It also includes the NCCE (operating system and communication ports), since intrusions may not only occur over communication ports of the NCCE that are used by the DPE, but also over other ports of the NCCE. The latter point concerns mainly the administrative domains of end users (consumers) whose Costumer Premises Equipment (CPE), e.g., PCs or workstations, cannot be assumed to be exclusively used as the endpoint of the TINA network.
Service Security is mainly concerned with the preservation of the integrity

of service control. Service control includes, among others, the verification of whether a user is allowed to use a service (subscription) and the accounting for billing purposes. Both rely on the authenticated identity of the user. This must be supported by a protocol for authentication of the user. The integrity of service control includes integrity of subscription verification and accounting. Access to the service functionality is controlled at two levels, the DPE level and the service level. At the DPE level, a coarse-grained access control based on the authenticated identities of the users involved in a session prevents attempts by others to invoke operations of the service components involved in the session. At the service level, the service logic implemented in the service component controls the access to service specific information and functionality based on the authenticated identities, context, and state information (authorization). Integrity and confidentiality of the messages exchanged between the service components via operational interfaces must be achieved by the activation of the appropriate features of the DPE security services.

DPE Security is mainly concerned with the prevention of illegal access to service components as well as the protection of messages containing arguments, results, and exceptions of object invocations and notifications. DPE node security also provides the means to audit and report security relevant events on the node according to the audit specifications defined by the administrator. DPE security includes the security of the DPE implementation and its basic services.

Communication Contents Security is concerned with the authenticity, integrity, and confidentiality of the service contents information. Since all service content information in TINA is delivered in the form of streams, it deals only with streams, which are protected using cryptographic mechanisms, preferably stream ciphers or special ciphers for certain information formats (e.g., voice or video data). If the service implemented in the provider's domain does not require any modification of the stream between two users, they can have end-to-end security. Otherwise, only security between the provider and the respective users can be achieved. The management of the necessary keys is part of the service control.

Allocation of Security Functionality:

In TINA, each stakeholder in the network has its own administrative domain. We make the assumption that the administrative domain is the trust domain of the stakeholder. This assumption is based on the fact that in the regular case the installed hardware is under the physical control of the stakeholder and the software is installed by the stakeholder. Thus, we assume complete trust between the stakeholder and its current administrative domain. Security within the administrative domain (intradomain security) is domain specific and is achieved by local means (e.g., operating system security measures). Within its own domain, the stakeholder trusts in the correctness of the installed software. Towards the outside, the administrative domain must be protected against illegitimate access. For interactions with other domains (interdomain interactions), limited trust relationships must be established. The communication channels between domains cannot be assumed to be secure. Protection must be achieved by cryptographic means. Thus, we allocate the necessary security functionality and the responsibility for its administration to each domain and the corresponding stakeholder. This decision must be supported by an appropriate scheme of security identities. The security identities have to be the identities of the stakeholders (their domains), which are specified in the TINA naming framework. All objects in a domain must act to the outside under the identity of their domain. From the security point of view, the objects in one domain are seen by the outside as one entity. This assumption is reasonable, since we have only one stakeholder in each domain.

Since there is no central security administration, it must be assumed that each stakeholder has its own security policy. This heterogeneity includes also different preferences for cryptographic algorithms and protocols to achieve the same security goal. Thus, for each interaction between domains, a security context has to be negotiated between the stakeholders. Such a negotiation must be supported by the security architecture and the middleware. In order to be generally applicable but also controllable by the applications, the security functionality has to be independent from the applications above and the supplier specific NCCE below. Thus, the natural vertical allocation is the DPE. That means, it has to be aligned with the DPE platform (CORBA).

CORBA Security:
CORBA provides some security functionality, which we strive to use whenever possible in our security architecture for TINA. The CORBA Security specification (OMG 1995_2) has been released by the OMG to provide the model and the architecture for security in CORBA systems. Besides the general model it specifies the security facilities and interfaces available to application developers, security administrators, and implementors of secure CORBA systems. It touches on the problem of secure interoperability between different CORBA implementations, although this issue is discussed in more detail by a companion document, the Common Secure Interoperability specification (OMG 1996). The security functionality (i.e., authentication, access control, message protection, auditing, etc.) is provided by ORB security services that may rely on some underlying security technology, which itself may use operating system mechanisms and additional security hardware (Figure 3). During secure object invocations, the ORB intercepts the requests and replies between the client and the target object and calls the appropriate security services. Some of the security services can be invoked by the applications directly, to enforce their own security preferences. An important aspect of this architecture is that the components that implement the security services are independent of any specific security technology. The specification allows the use of an isolat-

ing interface (e.g., GSS-API (IETF 1993)) between the security services and the security technology. This enables the accommodation of different security technologies within the architecture, such as technologies based on operating system protection mechanisms, existing security components (e.g., cryptographic libraries), or a set of distributed security services.

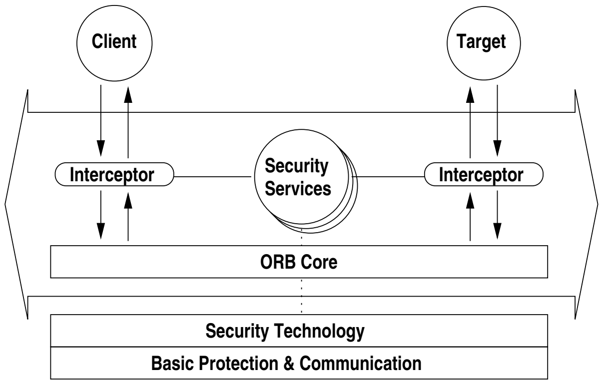

Figure 3 Model of Security in CORBA

The specification of secure interoperability between different CORBA implementations (OMG 1996) (OMG 1997) extends the CORBA 2.0 standard. A new protocol, the Secure Inter-ORB Protocol (SECIOP) is specified, which enables secure interactions of clients and target objects that reside on different ORBs, as long as the same security technology is used on both sides. The information, which security technology an object supports is part of the interoperable object reference (IOR). Based on the information in the IOR, a security context acceptable for both sides can be established. The establishment of the security association and the protection of messages are controlled by security tokens that are added to the inter-ORB protocols. Key management is not explicitly dealt with in the CORBA Security specification. Self-contained security technologies, including their protocols, are proposed in (OMG 1996) and (OMG 1997).

4 THE CRYSTINA SECURITY ARCHITECTURE

Security features in TINA are implemented at various levels. In our approach, the DPE offers general security functionality to the applications on each DPE node as part of the DPE functionality. To ease integration in applications, as much functionality as possible should be provided as self-contained DPE security services. However, the DPE should also provide lower level DPE security mechanisms to the applications for handling application specific security tasks.

The layering of the security functionality is illustrated in Figure 4. The DPE security services are exclusively based on the DPE security mechanisms. The implementation of these mechanisms may directly use cryptographic mechanisms or may be built on available higher level security technology, such as Kerberos (Neuman and Ts'o 1994). The underlying security technology may use the same cryptographic mechanisms as the DPE security mechanisms or proprietary implementations. The use of cryptographic mechanisms and/or higher level security technology may be accomplished through standardized interfaces (e.g., GSS-API) to facilitate the integration of existing products into the DPE. Above the DPE level, there are special security services, which rely exclusively on the DPE security services and mechanisms. They are used by TINA applications, but are applications themselves. The special security services are not implemented on each DPE node. Examples are electronic cash support or notary services.

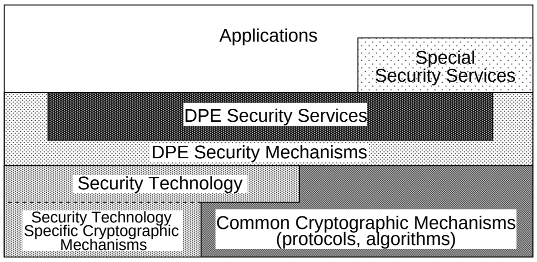

Figure 4 Layering of CrySTINA security features

Secure interoperability provided by CORBA security may not be sufficient for the TINA architecture. Secure interoperation between objects depends on their membership to security policy domains, security technology domains, and ORB technology domains. We assume that each TINA administrative domain is mapped onto one security policy domain and one ORB technology domain, and that each boundary between TINA administrative domains is also a boundary between security technology domains. The latter reflects that stakeholders with various kinds of customer premises equipment, varying priorities regarding security, and under possibly different national laws cannot be assumed to have the same security technology. Interoperability between objects in different security policy domains can, thus, only be achieved if both domains agree on a common security policy for the respective interactions. This common security policy can be negotiated at invocation time or in advance. Objects in different ORB technology domains can interact securely using the SECIOP, as long as the same security technology is used at both sides. According to the CORBA Security specification, interaction of objects in different security technology domains (e.g., objects belonging to different stakeholders in different countries) requires a security technology gateway. However, this may cause a trust problem, because such a gateway cannot be realized without the administrators of both security domains trusting each other or a third party that runs the gateway. A less restrictive solution that is not supported by CORBA would be to negotiate the security technology.

Below, in Figure 5, we present our model of security for distributed object based telecommunication architectures, such as TINA. It is based on the CORBA Security specification with some modifications according to the observations discussed above.

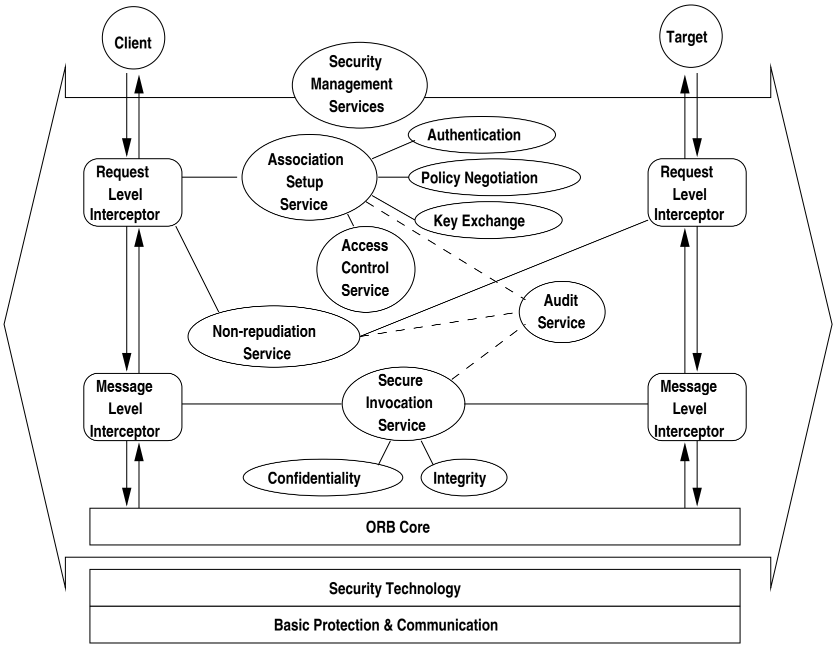

Figure 5 CrySTINA Model of Security

When a client invokes an operation on a target object, a request and in most cases a reply are passed between them. According to the CORBA Security specification and based on the observation of various CORBA implementations, we assume that the request and the reply can be intercepted at two levels: at the request level, where we have access to the request and the reply as structured data, and at the message level, where the request and the reply are available as an unstructured buffer containing the respective messages in a serialized form. These two levels of interception are very well adapted to support the enforcement of security, since some of the security services (e.g., access control) can best be performed on structured requests where the information about the involved principals and the requested operation is directly available, while other security functions (e.g., encryption) can more naturally be applied to unstructured raw data.

Each request is intercepted by the request level interceptor at the client side. If there is no security association established between this client and target object, then the Security Association Setup Service (SASS) is called and a security association is established between them. This means mutual authentication, security policy negotiation, and exchange of security related parameters (e.g., cryptographic keys, initialization vectors, etc.). The SASS uses the identity information of the stakeholder that owns the client in the authentication process. Based on the authenticated identity of the client, access control on the target object can be performed in this phase. If the access is not allowed, then the association is not established at all, and the client is notified (e.g, an exception is raised). An established security association is represented by security context information on both sides. Then the request is processed further according to the negotiated security policy (e.g., the Non-repudiation Service is called, if it is mandated).

On its way to the network, the request is intercepted by the message level interceptor as well, which calls the Secure Invocation Service. According to the negotiated security policy, integrity and/or confidentiality protection is applied using the security context information established before. The request is then passed to the ORB Core, which transfers it to the target side. At the target side, the applied services are called in reverse order (with the exception of the SASS, since the association has already been established).

Each service can call the Audit Service, if an event occurs that should be audited (e.g., an integrity violation has been detected). The placement of Security Management Services at the edge of the ORB system indicates that these services are usually called by management applications on behalf of the security administrator.

5 CONCLUSION

We introduced the TINA-C architecture, an open architecture for telecommunication services ranging from teleconferencing over video-on-demand to electronic commerce. We provided an analysis of the security problem domain in this architecture. As a result of this analysis, we allocated the necessary security functionality to each administrative domain and within the domain to the middleware layer (DPE), which is basically provided by CORBA products. CrySTINA, our security architecture for TINA, was presented. Because of its allocation to the middleware layer, the implementation of our architecture is closely related to CORBA security. Unlike the CORBA security architecture, CrySTINA can cope with the heterogeneity of security policies and security technologies, which must be expected as a side effect of the self-administration of the administrative domains in TINA. This is achieved by the negotiation of security contexts. Prototypical implementations of the concept use a commercial ORB product (Orbix 1997) and a free CORBA implementation in Java called JacORB (Brose 1997).

ACKNOWLEDGEMENT

This work has been supported by the Swiss National Science Foundation as part of the Swiss Priority Programme Information and Communications Structures (SPP-ICS) under project number 5003-045364.

REFERENCES

Barr, W.J. Boyd, T. and Inoue, Y. (1993) The TINA Initiative. *IEEE Communications Magazine*, March 1993, 70-76.

Brose, G. (1997) JacORB: Implementation and Design of a Java ORB. *proc. DAIS'97, IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems*, Cottbus, Germany, September 1997, Chapman & Hall.

Buttyán, L. Staamann, S. and Wilhelm, U. (1998) CrySTINA: Security in the Telecommunications Information Networking Architecture. Technical Report No. 98/259, École Polytechnique Fédérale de Lausanne (EPFL), Département d'Informatique (DI), January 1998.

Dupuy, F. Nilsson, G. and Inoue, Y. (1995) The TINA Consortium: Toward Networking Telecommunications Information Services. *IEEE Communications Magazine*, November 1995, 78-83.

Garrahan, J.J. Russo, P.A. Kitami, K. and Kung, R. (1993) Intelligent Network Overview. *IEEE Communications Magazine*, March 1993, 30-36.

Internet RFC 1508 (1993) Generic Security Service - Applications Program Interface (GSS-API).

IONA Technologies (1997) Orbix 2.1.

Magedanz, T. and Popescu-Zeletin R. (1996) *Intelligent Networks*. International Thomson Computer Press, London, 1996.

Netscape (1996) Secure Socket Layer (SSL).

Neuman, C. and Ts'o, T. (1994) Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, September 1994, 33-38.

Object Management Group (1995) The Common Object Request Broker, Architecture and Specification, Revision 2.0.

Object Management Group (1995) CORBA Security.

Object Management Group (1996) Common Secure Interoperability (CSI).

Object Management Group (1997) CORBAsecurity/SSL Interoperability.

TINA Consortium (1997) Service Architecture, Version 5.0.

BIOGRAPHIES

Sebastian Staamann is a researcher at the Operating Systems Laboratory (LSE) in the Computer Science Department (DI) of the Swiss Federal Institute of Technology in Lausanne (EPFL), and he chairs the special interest group on the security architecture of the TINA Consortium. He received a diploma in electrical engineering from the Technical University Berlin (TUB) and is currently finishing his doctoral dissertation in computer science. From autumn 1992 until spring 1997 he tought information and communication security at TUB. At EPFL he is conducting the CrySTINA project and two other research projects concerned with security in middleware based service architectures. His research interests are security and privacy issues in computer and telecommunications networks, cryptography, electronic commerce, and reliability in distributed systems.

Levente Buttyán is a research and teaching assistant, and PhD student at the Institute for computer Communications and Applications (ICA) in the Computer Science Department (DI) of the Swiss Federal Institute of Technology in Lausanne (EPFL). He received the M.Sc. degree in computer science from the Technical University of Budapest (TUB) in 1995. Before joining the EPFL, he spent one and a half year as a postgraduate student at the Department of Telecommunications at TUB, where he was concerned with the design issues of conventional block ciphers. His main areas of interest are cryptography and security in distributed systems. His current research topics are logic based design of authentication protocols and security for middleware based telecommunication service architectures (CrySTINA project).

Uwe Wilhelm is an assistant and PhD student at the Operating Systems Laboratory (LSE) in the Computer Science Department (DI) of the Swiss Federal Institute of Technology in Lausanne (EPFL). He received a diploma in computer science from the University of Kaiserslautern, Germany, in 1992, after which he spent a year as researcher at Cornell University, USA. Before joining the EPFL he spent several months as a private consultant for IDS. His main areas of interest are privacy, data protection, and security in computer and telecommunications networks as well as reliability in distributed systems. His current work is centered on security, which he explores in the use of trusted hardware for the protection of mobile agents and CrySTINA.