1 Multilateral Security in Middleware-Based Telecommunication Architectures

Levente Buttyán, Sebastian Staamann, Uwe Wilhelm

Swiss Federal Institute of Technology, Computer Science Department
EPFL-DI, CH-1015 Lausanne, Switzerland
{Levente.Buttyan, Sebastian.Staamann, Uwe.Wilhelm}@epfl.ch

1.1 Abstract

The concept of using middleware-based architectures for telecommunication services in the broadband, multimedia, and information era is increasingly gaining acceptance. One representative example is the Telecommunications Information Networking Architecture (TINA), which is characterised by 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 customer premises equipment. Concepts for its security architecture are developed in the CrySTINA (Cryptographically Secured TINA) project. We introduce TINA, 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, which must be expected in TINA networks as a result of following the approach of multilateral security for global networks in a consistent way. Thus, we present a model for the enforcement of security policies that supports the negotiation of security contexts.

1.2 Introduction

Several recent publications discuss the state-of-the-art of the expected convergence of telecommunications and computing towards networking models for integrated services and applications (e.g., [3]). This phenomenon is witnessed by two trends. First, computers are increasingly used for the control of telecommunication systems, which enables more flexibility in control as well as the fast and cheap introduction of new telecommunication services. The basic idea stems from the Intelligent Network (IN) [6, 11] and is fully realised in the Telecommunications Information Networking Architecture (TINA) [4]. TINA defines an open architecture for telecommunication services implemented as distributed applications over a Distributed Processing Environment (DPE). The DPE is a software layer that offers a uniform interface to the underlying heterogeneous distributed system. It is expected to be based on products conforming to the Common Object Request Broker Architecture (CORBA) [13]. Second, with the Internet becoming a ubiquitous telecommunication infrastructure, providing telecommunication services over computer networks to business and residential end-users is becoming an attractive business area. Many new specialised service providers have developed service architectures for the Internet, such as Amazon [1] in the area of online book selling or Pointcast [18] in the area of information dissemination. In most cases, these proprietary service architectures are based on World Wide Web (WWW) technology.

None of the aforementioned approaches has gained exclusive acceptance by the telecommunication community. Instead, many telecommunication operators (e.g., British Telecom [22]) as well as some telecommunication equipment suppliers (e.g., Siemens [21]) have started to develop CORBA-based service architectures for service provision over the Internet, which use WWW technology as the means for the initial access to services by the end-user, and, potentially, Java applets as the glue between WWW and CORBA. The Java applets realise the end-user system part of the CORBA-based distributed application, which can be downloaded from the provider's service centre and executed on nearly all types of systems at the end-user's side. In most cases, these service architectures can be viewed as simplified versions of the TINA service architecture [25] tailored to the Internet.

These trends reflect several developments, which cause security problems that are new to the traditional telecommunications world. The diminishing cost of transmission bandwidth enables distributed multimedia real-time applications. The powerful Customer Premises Equipment (CPE) available due to the progressive use of computing technology on the user's side enables a multitude of services to be delivered via a common telecommunication infrastructure to multipurpose end-user terminals. The worldwide deregulation of the telecommunication environment creates an open market for the provision of telecommunication services. Thus, telecommunication networks are not only populated by a multitude of users but also by a multitude of service providers. The co-operation and, at the same time, competition of various providers in the same physical network, as well as the expected significance of the network acting as the infrastructure for services varying from teleconferencing and video-on-demand to electronic commerce and electronic banking raise a strong demand for multilateral security and privacy of service usage and communication. An important issue with regard to multilateral security is the expected heterogeneity of security policies and security technologies of various network operators, service providers, and end-users.

In the CrySTINA project, we develop a security architecture for middleware-based telecommunication service architectures, such as TINA, that is aligned with the CORBA security architecture [14], but additionally supports the negotiation of security contexts in order to ensure secure interaction between domains with different security policies (multilateral security). In this article, we present this architecture and the rationale behind it. In Section 3 we introduce TINA as a reference architecture for middleware-based service architectures. TINA is currently the most encompassing effort to define an open architecture for telecommunication services implemented as distributed applications. As such, it can be regarded as the superset of most other middleware-based service architectures. Hence, the CrySTINA security architecture developed for TINA can easily be adapted to other middleware-based service architectures as well. Before presenting this security architecture in Section 5, we analyse TINA with regard to security in Section 4. Familiarity with TINA is not required, while knowledge about CORBA and security concepts is assumed.

1.3 The Telecommunications Information Networking Architecture (TINA)

TINA has a general business model with the stakeholder roles identified in Figure 1. Consumers buy services from Retailers, but the actual services are provided by Third Party Service Providers. Connectivity Providers offer the necessary connectivity (streams) for the transport of content information between stakeholders. The Broker acts as a kind of yellow page and white page service (i.e., it delivers references for services that can be described by service characteristics but also by names of providers).

Each stakeholder has its own administrative domain and can act in one or more roles. Any arbitrarily complex relationship for service use 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 and reliable temporary relationship between the user domain and the provider domain that is a prerequisite for usage interactions.

Interoperability between domains is guaranteed by the definition of inter-domain reference points. Figure 1 shows the business model with the defined roles and reference points.

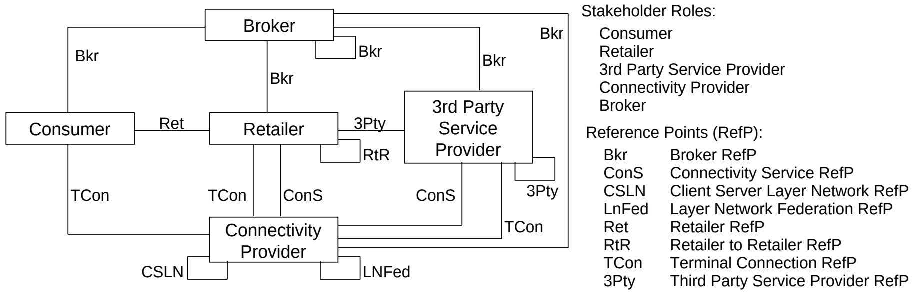

Figure 1: TINA business model

In TINA, services are realised as distributed applications (Figure 2). 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 is expected to consist of CORBA implementations as the DPE kernel and additional TINA specific services.

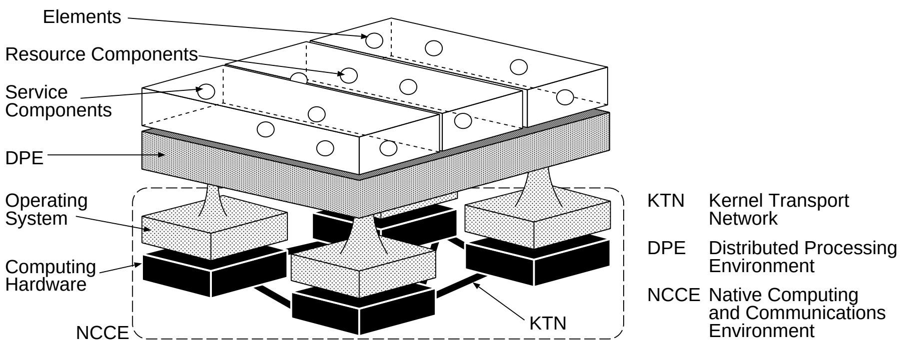

Figure 2: TINA overall architecture

Service components are specified in the TINA Object Description Language (ODL), which is a superset of the CORBA Interface Description Language

(IDL). They consist of computational objects (CO) 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 (which is not depicted in Figure 2).

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 are concerned with service usage). Communication sessions controlled by service sessions are responsible for establishing and maintaining the necessary stream connections for content delivery. The service session represents the instances of service usages and the state of the service logic. Exactly one provider is involved in a service session, but one or more users may 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 multi-user 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 3 shows the service components in their administrative domains and their relation to different sessions using the example of a videoconference with two participants. For a full description of all service components, we refer to [25] or [24].

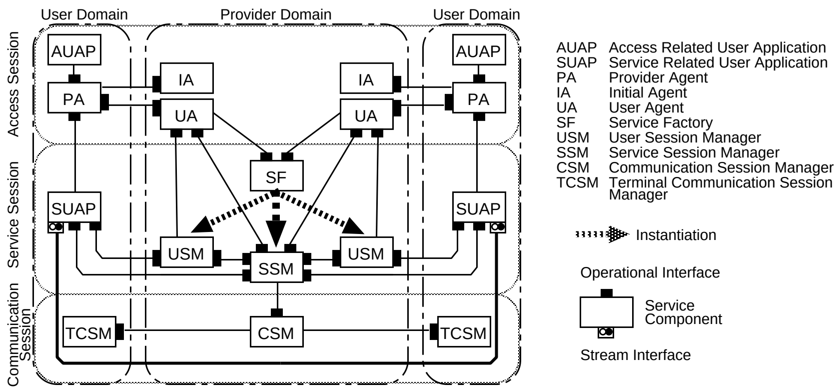

Figure 3: TINA service architecture

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 its UA (i.e., to establish 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 is then performed between the PA and the IA as part of the establishment of the access session. After the authentication, a reference to the UA is delivered to the PA. Additionally, a security association for further interactions between service session related service components is established.

1.4 Security analysis

Security concerns all parts of a TINA system; it is pervasive and cannot be addressed in isolation. In order to cope with this complexity, it is necessary to structure the security problem domain in an appropriate way. All services and resources may be the target of attacks. Attacks may be the illegitimate use of components or the modification of data, state, or programs. They may occur through direct access to systems, data, or services from outside or through modification of messages exchanged between interacting components. Potential attackers are outsiders, but also other stakeholders in the TINA network. Motives of attackers may be the illegitimate use of services, fraud (in online businesses as well as with regard to the charges for service use), eavesdropping on and observation of consumers or providers, or the deliberate prevention of service provision (denial of service attack). The ultimate goal of an attack may be achieved directly or indirectly. In the latter case, an attacker may install a back door during a first successful attack, which enables him later on (and possibly at multiple times) to carry out the actually intended misuse.

1.4.1 Structuring

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 identify the following sub-domains 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 (end user equipment will often be used for multiple purposes).
- **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 accounting for billing purposes. Both rely on the authenticated identity of the user, which must be supported by an appropriate protocol. 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 (authorisation). 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 transmitted 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. Since our architectural placement of security functionality allocates the general security services and mechanisms to the DPE (see Section 5), also the security of the security services themselves is part of DPE security.
- **Communication Contents Security** is concerned with the authenticity, integrity, and confidentiality of the service contents information, which in TINA is exclusively delivered in the form of streams. These streams are protected using cryptographic mechanisms, preferably stream ciphers [19, 20] 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 user-provider security can be provided. The management of the necessary keys is part of the service control.

1.4.2 Allocation of Responsibility

The most important criterions for the horizontal allocation of security functionality are who administers a domain, what security functionality is installed in the domain, and who has the physical control over both. 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. However, it is also possible that not all parts of the software implementing a service are pre-installed at the stakeholder's site. Some parts of the service implementation might be downloaded from another stakeholder (e.g., user interfaces, middleware functionality, and security modules that shall be run on the user's side might be downloaded from the provider). The authenticity and integrity of this mobile code must be guaranteed by additional means. A solution for this problem can be based on the concept of *signed applets* provided by Java product suppliers and its integration into the most popular web browsers. Thus, we assume complete trust between the stakeholder and its current administrative domain.

Security within the administrative domain (intra-domain 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 (or downloaded) software. Towards the outside, the administrative domain must be protected against illegitimate access. For interactions with other domains, 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 its corresponding stakeholder. This decision must be supported by an appropriate scheme of security identities, which have to be the identities of the stakeholders (their domains) as 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 perceived from the outside as one entity. This assumption is reasonable, since we have only one user (the 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. That means, it has to be aligned with the DPE implementation (i.e., CORBA).

1.5 The CrySTINA approach

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. This layering of 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 [12]. 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 standardised interfaces (e.g., GSS-API [10]) to facilitate the integration of existing products into the DPE. Above the DPE level, there are special security services, which also 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.

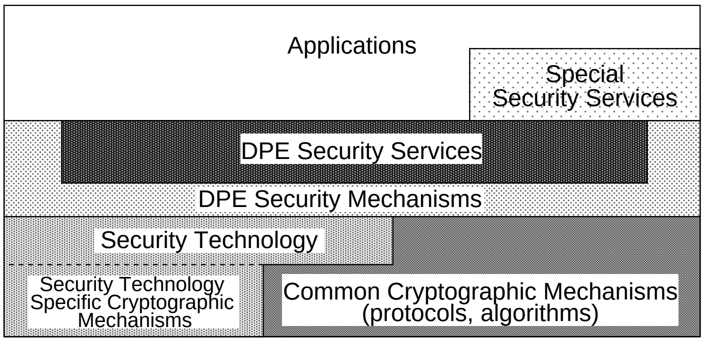

Figure 4: Layering of CrySTINA security features

1.5.1 CORBA security

Since the TINA DPE is provided by CORBA products, a natural starting point for the TINA security architecture is the CORBA Security specification [14]. It has been released by the OMG to provide the model and the architecture for security in CORBA systems. Its main concern is to ensure the well established security goals of confidentiality, integrity, accountability, and availability, which are achieved within the general model that specifies the security facilities and interfaces available to application developers, security administrators, and implementors of secure CORBA systems. Furthermore, 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 [15].

The security functionality 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 5). During secure object invocations, the ORB intercepts the requestes and replies between the client and the target object and calls the appropriate security services to perform identification, authentication, access control, auditing, etc. Some of the security services can be invoked by the applications directly, to enforce their own security preferences (e.g., non-repudiation). In many cases, objects perform operations on behalf of the initiator of a chain of object invocations, where the initiator needs to delegate some or all of its privilege attributes to the intermediate objects that will act on its behalf. The CORBA Security specification is very general and enables virtually all kinds of delegation models (simple, composite, combined and traced delegation).

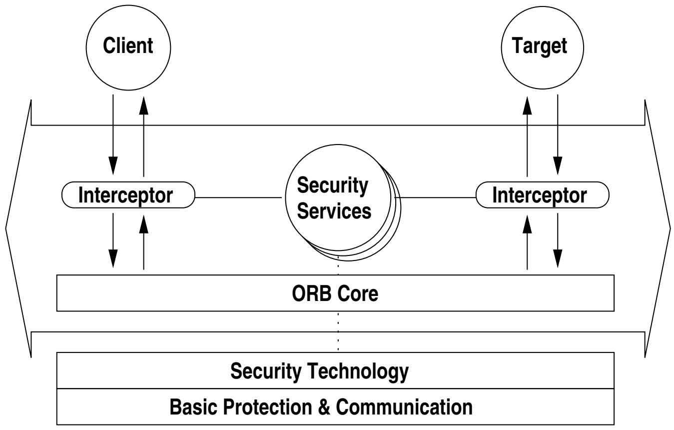

Figure 5: Model of security in CORBA

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 isolating interface (e.g., GSS-API [10]) between the security services and the security technology. This enables the accomodation of different security technologies based on operating system protection mechanisms, existing security components (e.g., cryptographic libraries), or a set of distributed security services.

A distributed object system may consist of a huge amount of objects with a possibly even larger amount of security associations between them. This fact raises the issue of scalability. In order to cope with the scalability problem, the notion of domains is introduced. Three types of domains with regard to security are defined by the CORBA Security specification: security policy domains, security environment domains, and security technology domains. A security policy domain is the scope over which a security policy is enforced. A security policy domain is administered by a single security authority. A security environment domain is a domain in which the enforcement of the security policy is achieved by local means (e.g., objects on the same machine). The security technology domain is a set of objects for which the same security technology (e.g., Kerberos [12]) is used to provide security.

Each security domain contains a security policy object that is the representation of a security policy (e.g., access control, delegation, secure invocation, or audit policy). Upon the creation of an object, the ORB implicitly associates the object with one or more security domains and then transparently enforces the security policies of those domains. In this way, security is provided to all applications, even to those that are unaware of it.

The specification of secure interoperability between different CORBA implementations, defined in the Common Secure Interoperability document [15], 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 is 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 [15] and [16].

1.5.2 Problems with CORBA Security

The generality of the CORBA Security specification makes it suitable to be the basis of security for a broad spectrum of business applications. In some respects, it is even more general than required, and it is questionable whether this rich functionality is necessary for certain families of applications, such as telecommunication services based on the TINA architecture. Let us consider, for instance, the access control schemes and delegation models described in the CORBA Security specification. TINA service components are implemented as TINA COs or CO groups. TINA COs may have multiple interfaces, as opposed to CORBA objects, which have exactly one interface. We assume that each interface of a TINA CO will be implemented by a dedicated CORBA object, and thus each TINA CO will be realised as a set of CORBA objects [9]. Since the functionality offered by a TINA CO is structured into interfaces according to the coherence of sub-functionalities, access control to a TINA CO can be applied at the granularity of TINA CO interfaces, which means that we only need to control access to whole CORBA objects. Furthermore, TINA service components always act on behalf of the stakeholder that owns them, therefore it is sufficient to support identity based access control schemes at the target side and no delegation is required.

For other aspects, secure interoperability provided by CORBA security may not be sufficient for the TINA architecture. Secure interoperation between objects depends on the membership of the objects 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 [23]. 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 technologies. 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 is 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 realised without the administrators of both security domains trusting each other or a third party that runs the gateway. A solution more oriented to multilateral security, which is not supported by CORBA yet, is to negotiate the security technology, as well.

1.5.3 The CrySTINA security architecture

In Figure 6, we present our model of security for middleware-based telecommunication architectures, such as TINA, which is based on the CORBA Security specification with some modifications according to the observations above.

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 serialised 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 operation is directly available, while other security functions (e.g., encryption) can more naturally be applied to unstructured raw data.

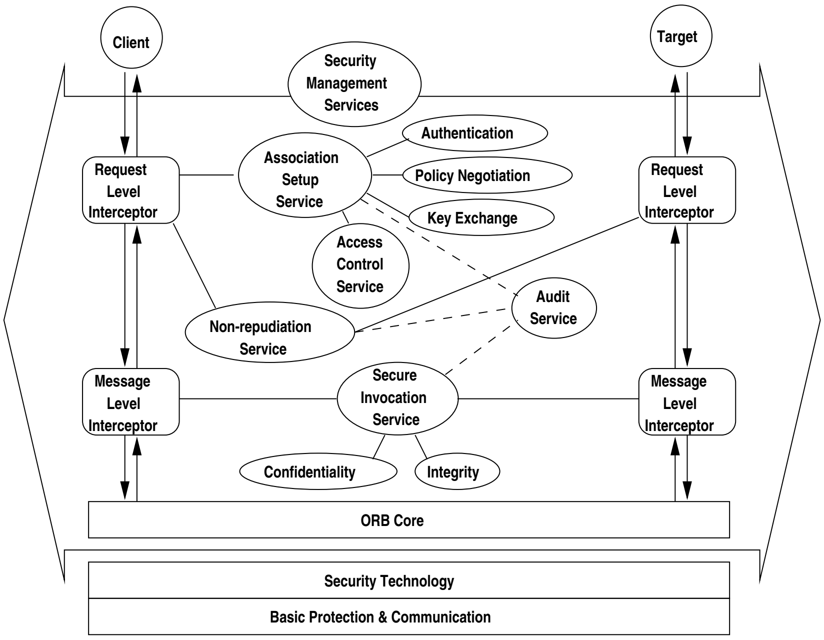

Figure 6: CrySTINA model of security

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 Association Setup Service 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, initialisation vectors, etc.). The Association Setup Service 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 is 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 same services are called in reverse order (with the exception of the Security Association Setup Service, 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.

The model described above can be implemented in several ways. One possibility is to implement the Security Association Setup Service, the Audit Service, and the Security Management Services as independent CORBA objects. These objects are unique in each domain and available to all applications in the domain. They provide interfaces

- to applications, through which they can request the Setup of new security associations, get established context information, and set security preferences that are not conflicting with the domain security policy,
- to other domains, through which they can authenticate each other, negotiate policies and exchange keys, and
- to management applications, through which they can manage credential and policy objects.

1.5.4 CrySTINA and the Internet

Considering the increasing importance of SSL [5] for Internet technology, we have also considered the use of SSL instead of SECIOP. If SSL is used as a secure transport layer between the ORB of the client and that of the target object, then some of the security services depicted on Figure 6 become unnecessary. Since SSL provides confidentiality and integrity protection of messages exchanged we do not need the Secure Invocation and the Key Exchange Services when using SSL. SSL can also provide mutual or unilateral authentication. If the client and the target sides both have X.509 [8] compliant public key certificates, then the SSL protocol can be used to mutually authenticate the client and the target object and we do not need the Authentication Service. If only the target side has a public key certificate, then the server authentication scheme of SSL can be used to authenticate the target object to the client. In this case, we need the Authentication Service of Figure 6 only for authenticating the client to the target object. However, even using SSL, we need the Policy Negotiation, the Access Control, the Non-repudiation, and the Audit Services.

1.6 Conclusion

We introduced the TINA-C architecture, which is an open architecture for a large variety of telecommunication services. 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. Ongoing work is concerned with the prototypical implementation of the concept using a commercial ORB product [7] and free CORBA implementations [2, 17].

1.7 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.

1.8 References

[1] Amazon online book-shop. *http://www.amazon.com/*

[2] Brose, G., 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.

[3] Decina, M. and Trecordi, V., Convergence of Telecommunications and Computing to Networking Models for Integrated Services and Applications. *Proceedings of the IEEE*, Special Issue on the Global Information Infrastructure, December 1997, 1887-1914.

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

[5] Freier, A.O., Karlton, P., and Kocher P.C., The SSL Protocol, Version 3.0. Internet Draft, *http://home.netscape.com/eng/ssl3/*

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

[7] IONA Technologies, Orbix online product information. *http://www.iona.com/products/orbix/*

[8] ISO/IEC 9594-8, Information Technology – Open Systems Interconnections – The Directory: Authentication Framework (Also ITU-T Recommendation X.509).

[9] Kitson, B., CORBA and TINA: The Architectural Relationships. *Proc. TINA'95 Conference*, Melbourne, Australia, February 1995, 371-386.

[10] Linn, J., Generic Security Service – Applications Program Interface (GSS-API) Internet RFC 1508.

[11] Magedanz, T. and Popescu-Zeletin R., *Intelligent Networks*. International Thomson Computer Press, London, 1996.

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

[13] Object Management Group, The Common Object Request Broker, Architecture and Specification, Revision 2.0., *http://www.omg.org/*

[14] Object Management Group, CORBA Security. *http://www.omg.org/*

[15] Object Management Group, Common Secure Interoperability (CSI). *http://www.omg.org/*

[16] Object Management Group, SSL/CORBA Security. *http://www.omg.org/*

[17] OOC Inc., ORBacus for C++ and Java. *http://www.ooc.com/ob/*

[18] PointCast Network. *http://www.pointcast.com/*

[19] Rueppel, R., *Analysis and Design of Stream Ciphers*. Springer, 1986.

[20] Schneier, B., *Applied Cryptography*, 2nd edition. Wiley, 1996.

[21] Schoenbauer, H., Zoernack, A., Hussmann, H., Sevcik, M., DeZen, G., Marsiglia, M., Ricagni, G., and Vezzoli, L., Value-added Internet: a pragmatic TINA-based path to the Internet and PSTN Integration. *Proc. TINA'97 Conference*, Santiago, Chile, November 1997, IEEE CS Press.

[22] Smith, C., Applying TINA-C Service Architecture to the Internet and Intranets. *Proc. TINA'97 Conference*, Santiago, Chile, November 1997, IEEE CS Press.

[23] Staamann, S., Buttyán, L., Hubaux, J-P., Schiper, A., and Wilhelm, U., Security in the Telecommunications Information Networking Architecture - the CrySTINA Approach. *Proc. TINA'97 Conference*, Santiago, Chile, November 1997, IEEE CS Press.

[24] Staamann, S., and Wilhelm, U., Cryptographic Protection of Connection Integrity with Interruption Detection in TINA., *IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems,* Cottbus, Germany, September 1997, Chapman & Hall.

[25] Telecommunications Information Networking Architecture Consortium, TINA-C Service Architecture 6.0. *http://www.tinac.com/*