# Overall Integrity of Service Control in TINA Networks

*S. Staamann*
*Swiss Federal Institute of Technology - Lausanne*
*EPFL-DI-LSE, CH-1015 Lausanne, Switzerland,*
*fon +41-21-693-5267, fax +41-21-693-6770, e-mail staa@lse.epfl.ch*

**Abstract**

This article is concerned with protecting service control against interruption attacks. It investigates threats to service control in the Telecommunications Information Networking Architecture. Traditional security services protect message flows, but not service logic. Attackers can still interrupt service control connections and take advantage of the caused influences on service control, even if the messages are cryptographically protected. The focus of this article is to prevent this kind of attack. A protocol for protecting overall integrity of service control by means of a cryptographic protocol is presented and its use is demonstrated.

**Keywords**

Integrity, service control, TINA, cryptographic protocols, security

## 1 INTRODUCTION

The beginning competition in the telecommunications market, the diminishing cost of transmission bandwidth, and the use of computers are changing the telecommunication environment. New architectures for telecommunications networks are under development. They shall enable competition of service providers, a big variety of services, and the fast and cheap introduction of new services. Security is becoming an essential requirement of network architectures. Current architectures for distributed computing and information networking such as DCE, CORBA, and TINA include security architectures which provide basic security services, such as authentication of users and data origin, access control, integrity and confidentiality of transmitted data, and non-repudiation.

Future multiservice and multiprovider telecommunications networks will pose new security problems. Not all of them can be addressed by using the traditional security services which are provided by DCE (X/Open 1994) and CORBA (OMG 1995_1). The increasing complexity of service control enables inside and outside attackers to take advantage of deliberately caused interruptions of message flows of service control interactions. Particularly, the flexibility of accounting and billing schemes and multiparty communications may attract such attacks on service control and cannot be prevented easily. In this article, we investigate threats to user security which may be caused by the interruption of service control. Our analysis is carried out on the TINA architecture. We present a cryptographic protocol for the protection of service control connections and demonstrate the use of this protocol. We developed the protocol for the TINA architecture, nevertheless, we think, it may also be applicable in other telecommunications architectures. In our opinion, TINA is the most flexible architecture and the most promising approach to future telecommunications networks. Thus, we believe, TINA is the best choice to scrutinize the overall integrity of service control for multiservice, multiprovider, and multimedia networks.

Section 2 introduces TINA and its relevant concepts[1]. No acquaintance with TINA is assumed, but some familiarity of the reader with the basic concepts of CORBA is expected. Section 3 presents the service control concepts in TINA and section 4 gives an overview over the current state in TINA security and foreseeable developments in this field. Subsequently, we demonstrate threats which are caused by service control interruptions at an example. In section 6 we propose a method for preventing this kind of attack and section 7 presents the cryptographic protocol for this purpose.

## 2 TINA ARCHITECTURE

The Telecommunications Information Networking Architecture (TINA) (Barr *et al.* 1993) is an effort to define an open architecture for telecommunications services 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, and telecommunications and computer suppliers. The architecture is based on distributed computing and object orientation, it comprises concepts from ODP, IN, TMN and CORBA. TINA is applicable to various networks, broadband or narrowband.

In TINA, services consist of service components. Each service component is realized as one or a number of software units, called computational objects (COs). Service components and COs interact with each other via a distributed processing environment (DPE). The DPE provides a software sub-layer that operates above a native computing and communications environment (NCCE). Whereas the NCCE is technology dependent, the DPE offers a uniform interface to the distributed environment. The DPE is provided by a DPE kernel and DPE services. It is widely expected that the DPE will consist of CORBA

[1]The introduction represents the state at the time of writing. Since the TINA specifications are still under development, there may be minor changes until the time of publishing.

(OMG 1995_2) implementations as the DPE kernel and additional generic servers, such as traders and name servers. TINA services, composed of service components, are distributed applications above the DPE.

Components in the application layer are divided into three categories - service components, resource components, and elements. Service components address the service logic, service access, and service management. Services can make use of common resources by interacting with resource components. The resource components are high level abstractions of available resources which enable usage and management of these resources in a technology independent way. Elements are software representations of individual resources, such as transmission equipment, switches, and computers. Figure 1 shows the layering into applications, DPE, NCCE, and computing resources as well as the structuring in component categories. More detailed descriptions of the TINA overall architecture (TINA-C 1995) can be found in (Dupuy *et al.* 1995) and (Nilsson *et al.* 1995).

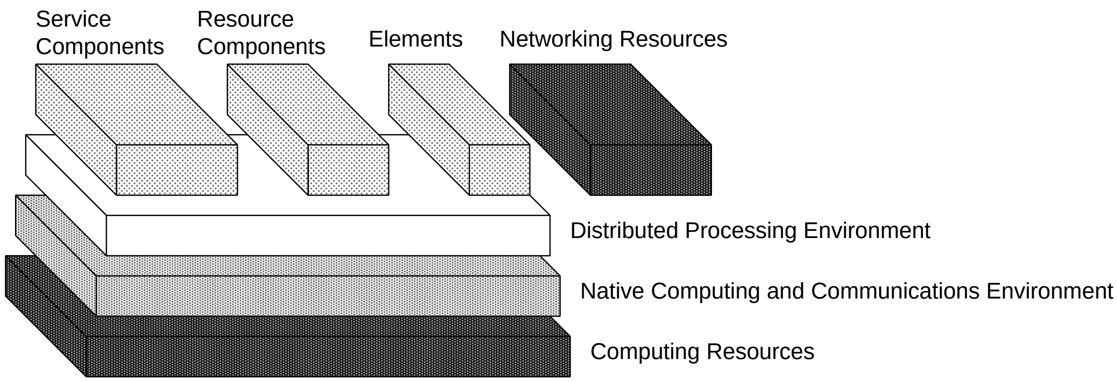


**Figure 1** TINA overall architecture

The TINA architecture is subdivided into four subsets: the Computing Architecture, the Service Architecture, the Management Architecture, and the Network Architecture. The Computing Architecture provides the concepts for how service components and computational objects are specified and how they interact. It defines the DPE as the computer platform support and provides the computational modelling concepts, such as the Object Description Language (ODL) which is a superset of CORBA's IDL. The main enhancements are the existence of multiple interfaces for one object, stream interfaces, and the grouping of objects. Operational interfaces are comparable to interfaces of CORBA objects. Stream interfaces do not have operations. The establishment of a stream between stream interfaces allows the passing of inherently structured information, e.g. audio and video bit streams. Figure 2 shows the graphical representations of objects and interfaces.

The business model of TINA is the supermarket. The following roles for stakeholders are identified: Consumer, Retailer, Third Party Service provider, Connectivity Provider, and Broker. Consumers buy services from Retailers, the services are provided by third Party Service Providers. Connectivity Providers offer the necessary connectivity (streams)

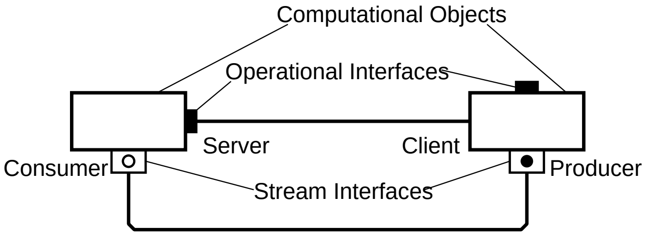


**Figure 2** TINA Computational Modelling Concepts

for the transport of content information between other stakeholders. The Broker acts as a kind of yellow page service, it delivers references for services which can be described by service characteristics or names of providers. Use of the Broker as a white page services is also considered. Each stakeholder has its own administrative domain. He can act in one or more roles. Complex relationships for service use are composed of two-party user-provider relationships. An example may illustrate this. Consumers A, B, and C have a video conference. They are all customers of Retailer X and get the video conference service from X. Retailer X does not provide the service himself, but has a wholesale agreement with a Third Party Service Provider Y who provides this service. In this example, there are four user-provider relationships: A-X, B-X, C-X, and X-Y. A user-provider relationship contains three types of interaction: access, ancillary usage, and primary usage. The access part is concerned with the establishment of a trusted and reliable temporary relationship between the user's current[2] domain and the provider. Such a relationship is a prerequisite for usage interactions. Primary usage is related to the actual service(s) the user has subscribed to. Ancillary usage assists the user in the contract with the provider, e.g. online subscription of primary services. Interoperability between domains is guaranteed by the definition of interdomain reference points.

## 3 SERVICE CONTROL IN TINA

The traditional call concept of telecommunications is in TINA substituted by the session concept. There are different kinds of sessions, according to the subdivisions in access and usage, service and communication links, and to the allocation to administrative domains. At the service level there are access sessions and service sessions. A service session is an instance of a service type. Exactly one provider is involved in a service session. One or more users participate in the service session. The service session consists of the provider service session with the global service logic in the provider domain and user service sessions. For each user, there is one user service session which is concerned with the user specific part of the service session. The user service session consists of two parts, one in

[2]TINA inherently supports personal mobility, i.e. the user can act in various domains as himself.

the provider domain and one in the user domain. A service session or a user service session, i.e. a user's participation in a service session, can be suspended and resumed later on, potentially from a different user domain (session mobility). 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. Two persistent sessions in their domains are involved in the access session: the user domain access session and the provider domain access session. The access session is established by interacting of these two sessions. Content information of services is in TINA passed in the form of streams. The communications session establishes and maintains the necessary stream connections for a service. It is controlled by the service session.

Sessions and other information objects are mapped onto service components. Service control is achieved by the interaction of service components in the users' and provider's administrative domains. Figure 3 illustrates the example of a conference with two participants. It shows the service components in their administrative domains and their relation to different sessions. Note that the figure does not actually show single computational interfaces but groups of them. In the following, we will briefly explain the service components and their interactions. For the sake of brevity, some aspects are omitted or simplified.

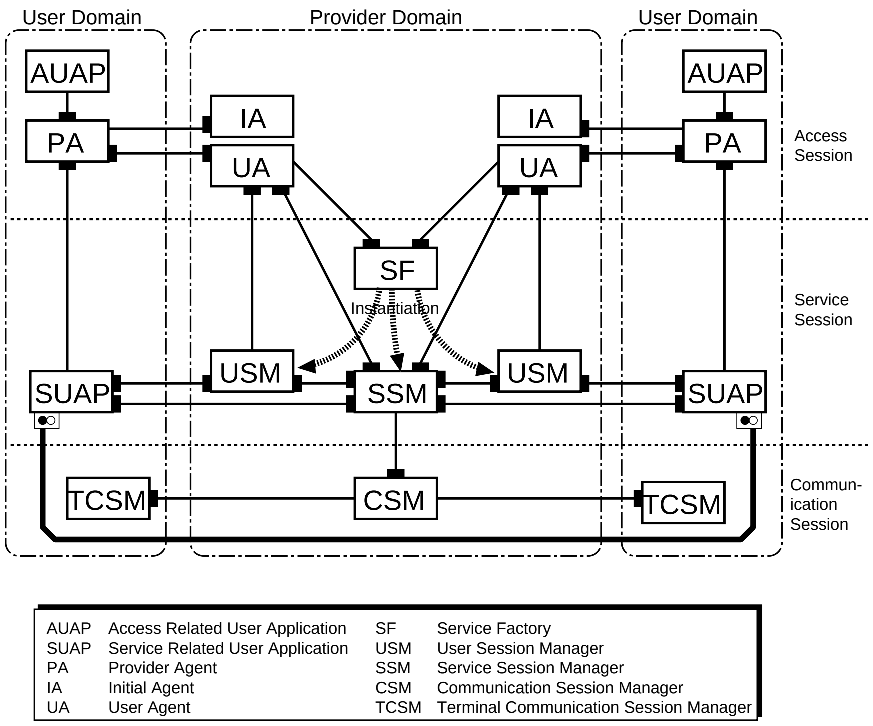


**Figure 3** TINA Service Components

The access related User Application (AUAP), the Provider Agent (PA), Initial Agent (IA), and User Agent (UA) belong to the access session. AUAP models applications in the user domain which support the access session. It provides the user interface for this purpose. The PA is the essential embodiment of the user domain access session, the provider domain access session is mapped onto IA and UA. AUAP, PA, IA, and UA are service independent. The IA is also user independent. It is the initial contact point for the users of the provider. For establishing an access session, the PA in the user domain contacts the IA in the provider domain. After the authentication, a reference to the UA is delivered to the PA. The UA represents the user in the provider domain. It is the user's contact point to start or resume a service session or the user's participation in a service session. The lifespan of the UA is the time of subscription of the user at the provider, the IA exists as long as the provider is active in his business.

The service session is mapped onto the service related User Application (SUAP), Service Factory (SF), Service Session Manager (SSM), and User Session Manager (USM). The SUAP models applications in the user domain which support the service. It embodies the user service session in the user domain. The SUAP is the endpoint of content information flows (streams) and of service control in the user domain. It interacts with the USM and the SSM in the provider domain. The USM embodies the provider domain part of the user service session, the SSM the provider service session. The SSM contains the global service logic, the USM represents the user related aspects of the logic in the provider domain. Both are service specific, they support service specific and service generic control functions. Generic functions are start, suspend, resume, and end of service sessions or the user's participation. An example for a service specific function is the rewind function of a video-on-demand service. Of course, the control functions of a service, generic and specific ones, must be supported by the SUAP of the service in the user domain. Thus, the SUAP is also service specific. In contrast to the UA, SSM and USM have a short lifespan. The SSM exists as long as the service session, the USM as long as the user service session, i.e. as long as the user participates in the service session. Both are created by the SF on request of the UA. For each new service session, the SF creates the SSM and the USM for the initiating user. For each user who joins the existing service session, an additional USM is created by the SF. The Communication Session Manager (CSM) and the Terminal Communication Session Manager (TCSM) are concerned with the establishing of stream between service components, such as SUAP in the user domain and SSM and USM in the provider domain. The CSM is controlled by the SSM. The detailed specification of service related aspects in TINA, including service control, is given in (TINA-C 1996). A good, but not quite up-to-date, description can be found in (Yagi *et al.* 1995).

## 4 TINA SECURITY

From the consumer's point of view, security is mainly concerned with preserving integrity and confidentiality of transmitted information. The means for protecting transmissions over untrusted communication links are provided by cryptography. In TINA, all contents information is passed in the form of streams between producers and consumers of this information. Authenticity, integrity, and confidentiality of these information streams can be achieved as end-to-end-security between the communicating parties by using standard cryptographic mechanisms, such as message authentication codes and encipherment. For performance reasons, symmetric ciphers, block ciphers as well as stream ciphers[3], shall be used for encipherment. The problem is the negotiation of keys. TINA systems must enable secure communications in case of first contact. In this case of spontaneous communication, the communicating parties do not have a common key. Thus, a proper distribution of symmetric keys for end-to-end security should be supported by TINA.

From the system perspective, the most important aspects of security are the protection of the single administrative domains against intrusion, the authentication of users and providers, the correctness of service executions, and the prevention of unauthorized service usage. Authentication is done during the establishment of the access session. The authentication scheme and the trusted third party (or certification authority for schemes based on asymmetric cryptography) are negotiated and the authentication protocol is executed. As the result, both sides have a session key which can be used for all interactions between user and provider during the access session. The authentication scheme and the key management infrastructure for the use of asymmetric cryptography, can be based on concepts from the Directory Authentication Framework (ISO *et al.* 1992) such as the two-way authentication exchange, public-key certificates and the certification hierarchy. Both can be used not only in the access session but also for the streams mentioned in the previous paragraph. This approach also enables non-repudiation services which are fundamental for the legal binding of transactions. The distribution of public keys is carried out by Brokers together with address information.

Unauthorized service usage is prevented at the service level. Each request for a new user service session (a user wants to join an existing service session) or for a new service session (start of a service) is checked before actions take place. The UA checks the subscription information, whether the user has subscribed to this service and under which conditions the user may use the service. If the session already exists, the SSM checks additionally, whether the state of the service logic allows the user to join the service session. Similar authorization checks have to be done for suspending and resuming of service sessions and user service sessions. Protection against illegitimate access to single service components or computational objects is done by access control at the DPE level.

The correctness of service execution requires the authenticity and integrity of exchanged service control information, i.e. the information passed via operational interfaces. For user privacy, confidentiality of those information should also be provided. These requirements are fulfilled by the use of security services of the DPE. CORBA implementations which comply with the CORBA security specifications (OMG 1995_1) will provide the means to guarantee authenticity, confidentiality, and message integrity using the session key of the access session. Still open is the question how CORBA systems with different security technology can securely interoperate. This is crucial for TINA, since it is widely expected that the DPE kernel will be provided by one CORBA system in each TINA administrative

[3] There may be legal constraints to the choice of cryptosystems and keys, e.g. regulations may prescribe the use of escrowed keys.

domain, and the CORBA security specification allows various underlying security technologies, such as Kerberos and SESAME. In TINA, the use of security mechanisms and formats should be negotiated as part of access session establishment. Assumed the availability of the respective mechanisms as part of the security technology in both domains, interoperability can be reached this way.

## 5 THREATS TO SERVICE CONTROL BY INTERRUPTION

Obvious threats to service control are the masquerade of attackers as subscribed users and the modification of service control messages. Masquerade is prevented by authentication as described above. The modification of single service control messages is prevented by security services which detect modification of a message after receiving. This is done by using cryptographic mechanisms, such as message authentication codes. Neither in TINA nor in CORBA specifications, the granularity of integrity is mentioned. It can be assumed that parameters, results, and exceptions of operation invocations are the units for which integrity is guaranteed. This means that complete messages at this level (invocation with parameters, results, exceptions) can be subject to interception, replay, and reordering. Attackers can this way deliberately disturb service control to take advantage. One obvious instance of such an advantage is the illegitimate use of services at the cost of another user. The following, rather unconventional, example may illustrate this.

Home entertainment is expected to be one big business area in providing services in TINA networks. Many Retailers will offer video games to their Consumers. The games, among them multi user games, will be provided as services. For simplicity, in our example, we assume a multi user game service which is provided by the Retailer himself, i.e. no additional Third Party Service Provider is involved. One of main goals of TINA is the support of complete flexibility of accounting and billing. It is completely in the power of the provider to determine the accounting scheme. In our example, one sensible scheme would be that the user who creates the service session (starts the game) has the control over the global aspects of the service (chairing) and is accounted for the session as long as he chairs it. After starting the game, other players can join the session. A service specific operation allows to hand over accountability and chairing to an other player. In the case, the chair wants to resign and no other player takes over, the chair may break off the game.

Since the communication links between user domain and provider domain are not physically secure, other players can intercept or replay messages of a player. To improve his own chances to win, one player could impede another one in playing by intercepting messages. This kind of attack, although in our example anyway not a real concern, can be prevented by the use of integrity sequence numbers or cryptographic chaining in the process of generation of message authentication codes. These mechanisms prevent the loss, replay, and reordering of messages within a sequence, but they do not detect complete interruption. In our example, interruption can be used for accounting fraud, which is surely of greater importance than the attack mentioned above. Here is one scenario: The chair announces that he wants to resign. No other player takes over chair and accountability. Instead, one of the other players interrupts the link between chair and provider. As the result, the chair remains accountable for the service session, although he cannot influence it anymore. The other players can continue to use the service at his cost against his will. Complete interruption of a communications link is physically even easier than replay or reordering. In the case of a later legal dispute, the usufructuaries of the attack can hardly be made responsible.

One can imagine similar attacks on many other multi user services, such as conferences, joint editing etc. The general problem is that users keep staying in an accountable state, even when the control connection is interrupted. Additionally, for services which do not require frequent service control operations, such as video-on-demand, pay-tv etc., to save service charges, users may untruly claim that they were unwillingly kept in an accountable state because the control connection was interrupted.

From an even more general perspective, there can be made the distinction in secure and critical states of a service participation. An accountable state is just one instance of critical states. Another instance is the exposure of information to access by other users in the course of a service. An critical state is any state of a user service session or provider service session in which the user has a disadvantage additionally to the unavailability of the service when the connection between user domain and provider domain, more precisely between SUAP and USM respectively SSM, is interrupted.

# 6 HANDLING OF CRITICAL STATES

The threats which result from critical states can be minimized if the respective session passes from the critical state into a secure state when the service control connection is interrupted. Which state is considered secure and which critical is service dependent. It is the responsibility of the service designer to categorize states of sessions as secure or critical. Both parts of a service session are relevant for the analysis: the provider service session (one for each service session) and the user service session (one for each user in the service session). The state of the user service session is controlled by interactions between the SUAP in the user's domain and the USM. The state of the provider service session (the global view of the service session) is controlled by interactions between the SUAPs in the users' domains and the SSM.

Each session must be assessed by the service designer whether the session can be stuck in an critical state when interdomain control connections are interrupted. The result of this assessment is the list of critical states of this session. Each of these states is dependent on one or more relevant interdomain control connections. For each pair of critical state and relevant connection, the designer specifies a state into which the session passes when the relevant connection is interrupted. Critical states of user service sessions are only dependent on the USM-SUAP control connection, but the provider service session may be dependent on various SSM-SUAP connections. For the provider service session, there may be transitions to different states from the same critical state, since the participating users may act in different roles (in the multi user game example: chair and ordinary players) and the relevance of their control connections may differ.

To initiate the specified state transition for a pair of critical state and control connection, the logic in the respective service component (USM or SSM) must notice the interruption of the control connection. A prerequisite for that is the employment of a security mechanism which serves for the detection of connection interruption. According to the TINA design principles, this mechanism is implemented as part of an universally usable security service. The following section presents the cryptographic protocol for the protection of overall connection integrity, including the protection against interruption. Unfortunately, the use of this protocol causes some additional traffic between user and provider domain. Thus, the "protection against interruption" feature of the security service should not always be activated. It is switched on by the using service component (USM or SSM) before passing into a critical state and switched off after leaving this state. When the feature is activated and the interruption of the relevant connection is detected by the service, it triggers the specified state transition.

# 7 INTEGRITY OF SERVICE CONTROL CONNECTIONS

The only way to check whether a connection still exists is the exchange of probe messages. We demonstrated that it is important for service components in the provider domain to notice an interruption. We could not find any threat except the unavailability of service control functionality which would result if the user does not notice the interruption of the service control connection. Thus, in our protocol, probe messages are only sent from the user to the provider domain. For performance reasons, the protection of the connection itself is combined with message and sequence integrity in the same protocol. Sequence integrity of messages can be achieved by cryptographic chaining of consecutive messages or by the use of sequence numbers. In contrast to cryptographic chaining, the use of sequence numbers allows the receiver to tolerate the loss of one or more messages without loosing the chance to verify the subsequent messages. Thus, we decided in favour of the sequence number approach. Nevertheless, the protocol can easily be modified to use cryptographic chaining.

The protocol is simple. As a prerequisite, it requires that both parties have already a common session key for use in the service session. It is sensible that the same session key that was distributed to both parties as the result of the authentication during establishment of the access session is used. For protecting sequence integrity, both parties use sequence numbers. To each message, the sender adds a sequence number and generates a message authentication code (MAC) over both using the session key. The receiver checks the MAC and compares the current sequence number with the one of the previous message. If the difference is different to one, the receiver knows that the sequence integrity was violated. When the "protection against interruption" feature is activated by the service component in the provider domain, a control message at the security service level about this activation with a timeout_time as parameter is sent to the respective security service entity of the SUAP in the user domain. At the same time, the security service entity for the connection in the provider domain starts its clock and expects the next message. Always when a message has arrived, the clock is reset. If no message arrives within the timeout time since the last clock reset, the state transition is triggered. In the user's domain, after receiving the activation message, also a clock is started. If no real control message is sent within the timeout time, a probe message is sent. Probe messages contain only the sequence number, but are also integrity protected by adding a MAC. When a message is sent, regardless

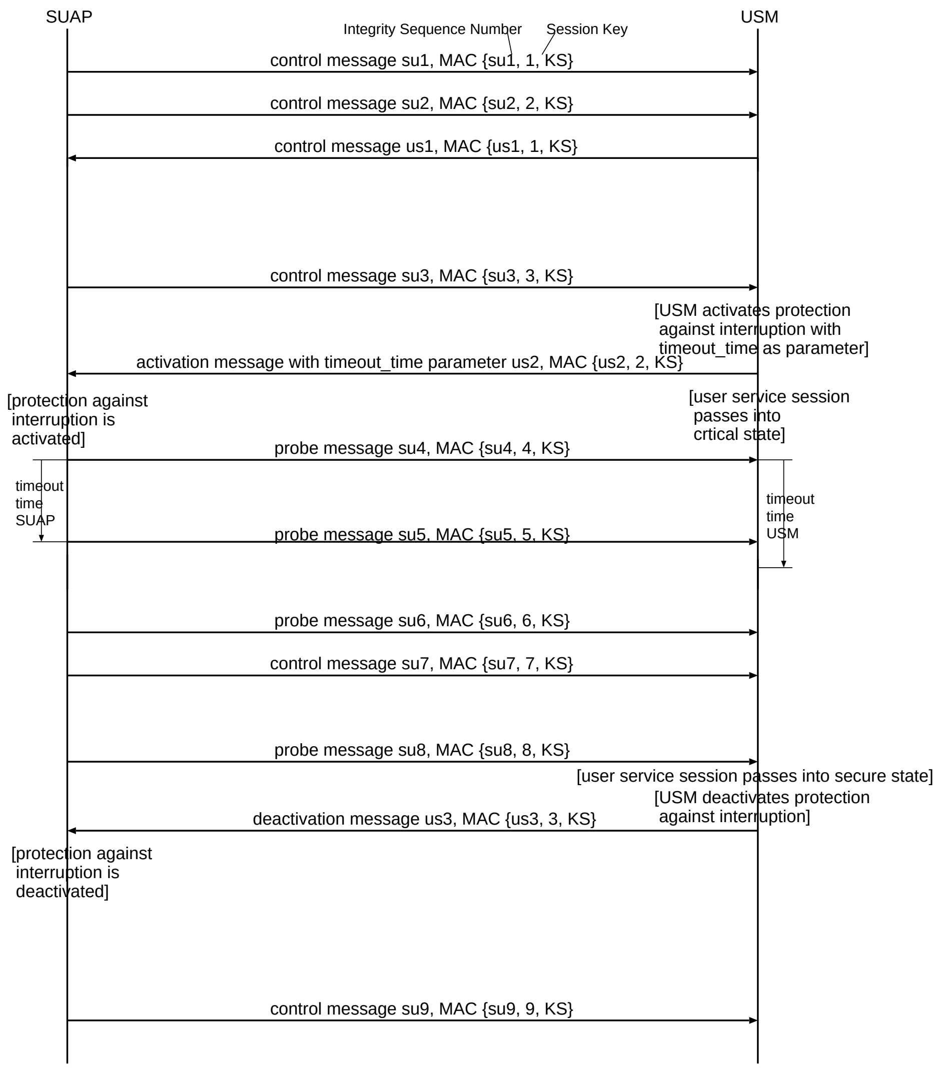


**Figure 4** Sample Message Transmission Event Trace

whether real or probe, the user domain clock is reset. Real messages and probe messages are numbered in the same sequence. When the "protection against interruption" feature is deactivated by the service component in the provider domain (the session has left the critical state), a control message at the security service level about this deactivation is sent to the user domain and the clock in the provider domain is halted. On receipt of the deactivation message in the user domain, the sending of probe messages is stopped. Figure 4 shows for the SUAP-USM instance the passing of messages at the security service level for an example period.

The timeout time in the user domain should always be chosen a bit less than the one in the provider domain for taking into account transmission time of control messages and processing time for probe message generation. The timeout time is a variable parameter. It is determined by the service designer and the service provider. Sensible values depend on the time sensitivity of the critical state, and the acceptable plus of traffic and the guaranteed response time in the control network. The protocol can also be used together with the "integrity with recovery" feature of security services. In this case, each retransmitted message gets an additional new sequence number. This way, also retransmission messages are used as message events for the protection against interruption and save probe messages.

## 8 CONCLUSION

The Telecommunications Information Networking Architecture is a flexible and promising architecture for future telecommunication networks. Nevertheless, the considerable flexibility of service control causes new threats of security which cannot be prevented by the use of established security mechanisms. It was demonstrated that the interruption of service control connections may cause damages, such as toll fraud. We proposed a method to prevent this kind of attack. A cryptographic protocol to be used in this method was introduced. Future work will be concerned with the realization of the cryptographic protocol as part of a secure and reliable DPE and the support of programming techniques for service components to use our approach.

## ACKNOWLEDGMENT

This work has been supported by the Swiss National Fund for Scientific Research as part of the Programme Prioritaire (PP-SIC) 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.

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

ISO/IEC Standard 9594-8 (1992) The Directory: Authentication Framework. (Also ITU-T Recommendation X.509).

Nilsson, G. Dupuy, F. and Chapman, M. (1995) An Overview of the Telecommunications Networking Architecture. *Proceedings of the TINA Conference 1995*, 1-12.

Object Management Group (1995) CORBA Security.

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

TINA-C Document TB_MDC.018_1.0_94 (1995) Overall Concepts and Principles of TINA.

TINA-C Document TB_RM.001_4.0_96 (1996) Service Architecture, Version 4.0.

X/Open Guide G410 (1994) Distributed Security Framework.

Yagi, H. Ohtsu, K. Kazuyuki, W. Wakano, M. Kobayashi, H. and Minerva, R. (1995) TINA-C Service Components. *Proceedings of the TINA Conference 1995*, 113-124.

## BIOGRAPHY

Sebastian Staamann is a research assistant at the Operating Systems Department of the Swiss Federal Institute of Technology in Lausanne. He received a diploma in electrical engineering from the Technical University Berlin and is currently finishing his doctoral dissertation in computer science. His research interests are security and privacy issues in computer and telecommunications networks, cryptography, and fault-tolerance in distributed systems.