ATIS STANDARD

# ATIS-1000655.2001(R2011)

# Signalling System Number 7 (SS7) – Upper Layer Security Capability

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS

ATIS is the leading technical planning and standards development organization committed to the rapid development of global, market-driven standards for the information, entertainment and communications industry. More than 250 companies actively formulate standards in ATIS' 18 Committees, covering issues including: IPTV, Service Oriented Networks, Energy Efficiency, IP-Based and Wireless Technologies, Quality of Service, and Billing and Operational Support. In addition, numerous Incubators, Focus and Exploratory Groups address emerging industry priorities including "Green", IP Downloadable Security, Next Generation Carrier Interconnect, IPv6 and Convergence.

ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications' Sectors, and a member of the Inter-American Telecommunication Commission (CITEL). For more information, please visit < http://www.atis.org >.

## AMERICAN NATIONAL STANDARD

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

## Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith.

ATIS-1000655.2001(R2011), *Signalling System Number 7 (SS7) – Upper Layer Security Capability*

Is an American National Standard developed by the **ATIS Packet Technologies and Systems Committee (PTSC)**.

*Published by*

**Alliance for Telecommunications Industry Solutions**
**1200 G Street, NW, Suite 500**
**Washington, DC 20005**

**ATIS-1000655.2001** (R2011)

(formerly T1.655-2001)

American National Standard for Telecommunications

# Signalling System Number 7 (SS7) – Upper Layer Security Capability

Secretariat

**Alliance for Telecommunications Industry Solutions**

Approved March 13, 2001

**American National Standards Institute, Inc.**

## Abstract

This standard describes the Security network capability, which allows an end user service in an originating Signalling Point (SP) to invoke various security functions in the originating and/or destination SP. The Security capability can be used for identification and authentication of the communicating entities; it also provides information that supports resource access control, system access control, and encryption and decryption functions.

## Foreword

The information contained in this Foreword is not part of this American National Standard (ANS) and has not been processed in accordance with ANSI's requirements for an ANS. As such, this Foreword may contain material that has not been subjected to public review or a consensus process. In addition, it does not contain requirements necessary for conformance to the standard.

This document is entitled the *American National Standard for Telecommunications – Signalling System Number 7 (SS7) – Upper Layer Security Capabilit*y. It is based on the Generic Upper Layer Security (GULS) functions described in *Information Technology - Open Systems Interconnection Upper Layers Security Mode*l, ISO/IEC IS 10745, June 1993. This standard is the result of work by members of the T1S1.3 Working Group on U.S. Standards for Common Channel Signalling. This revision to the standard includes the KeyExchange parameter, associated procedures, and informative annexes E and F, giving examples of exchanging encryption keys. Descriptions of parameters now included in T1.114-2000 have been removed.

This standard is intended for use in conjunction with *American National Standard for Telecommunications – Signalling System Number 7 (SS7) – Transaction Capabilities Application Part (TCAP),* T1.114-2000.

Future control of this document will reside with Accredited Standards Committee on Telecommunications, T1. This control of additions to the specification, such as ongoing protocol evolution, new applications, and operational requirements, will permit compatibility among U. S. networks. Such additions will be incorporated in an orderly manner with due consideration to the ITU-T layered model principles, conventions, and functional boundaries.

Suggestions for improvement of this standard will be welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, 1200 G Street NW, Suite 500, Washington, D.C. 20005.

This standard was processed and approved for submittal to ANSI by the Accredited Standards Committee on Telecommunications, T1. Committee approval of this standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the T1 Committee had the following members:

E.R. Hapeman, T1 Chair
W.R. Zeuch, T1 Vice-Chair
J.A. Crandall, T1 Director
S.M. Carioti, T1 Disciplines
S.D. Barclay, T1 Secretary
C.A. Underkoffler, T1 Chief Editor
W.B. Downum, T1S1 Technical Editor

**EXCHANGE CARRIERS**

| Organization Represented | Name of Representative |
|---|---|
| AT&T Wireless Services, Inc. | Peter Musgrove |
| Bell Atlantic | Josephine Gallagher<br>James F. Baskin (Alt.) |
| BellSouth Telecommunications Inc. | Malcolm Threlkeld, Jr.<br>John Spencer (Alt.) |
| Covad Communications Co. | Ron Marquardt<br>David Rosenstein (Alt.) |
| GTE Telephone Operations | Thomas Deaton<br>Gary E. McAninch (Alt.) |
| NorthPoint Communications, Inc. | Mark Peden<br>Mike Borsetti (Alt.) |
| Qwest | James L. Eitel<br>Richard Prince (Alt.) |

| Organization Represented | Name of Representative |
|---|---|
| Rhythms | Rand Kennedy<br>David Reilly (Alt.) |
| Rogers Wireless Inc. | Edward O'Leary<br>Peter Oldfield (Alt.) |
| SBC Communications, Inc. | C.C. Bailey<br>John E. Roquet (Alt.) |
| Sprint – Local Telecom. Division | Leroy D. Kellogg |
| US Telecom Association (USTA) | Paul Hart<br>Donald G. Bender (Alt.) |

**GENERAL INTEREST**

| Organization Represented | Name of Representative |
|---|---|
| Aerial Communications | George P. Lynch<br>Rob Rowe (Alt.) |
| AT&T Broadband | Paul Hughes<br>Jim Dahl (Alt.) |
| BellSouth Cellular Corp. | Don Zelmer<br>Andy Clegg (Alt.) |
| BOPS Inc. | Ali S. Sadri, PhD |
| CSI Telecommunications | Michael S. Newman<br>William J. Buckley (Alt.) |
| Catapult Communication | Katya Gircus<br>Nancy Gayed (Alt.) |
| Defense Information Systems Agency | Don Choi |
| Golden Bridge Technology Inc. | Kourosh Parsa<br>Karin Zickermann (Alt.) |
| Microcell Connexions | Venkatesh Sampath<br>Andrew Chow (Alt.) |
| National Communications System | Nicholas Andre<br>F. McClelland (Alt.) |
| NTIA | Neal B. Seitz |
| Pacific Bell Wireless | David Williams<br>Randolph Wohlert (Alt.) |
| Rural Utilities Service | Orren E. Cameron III<br>Norberto Esteves (Alt.) |
| Telcordia Technologies | Rick Harrison<br>Cliff Halevi (Alt.) |
| Voicestream Wireless Corp. | Gary K. Jones<br>Mark Younge (Alt.) |

**INTEREXCHANGE CARRIERS**

| Organization Represented | Name of Representative |
|---|---|
| AT&T | Doris S. Lebovits<br>Rick Canaday (Al |
| Bell Canada | P. Norman Smith |
| General Communication, Inc. | Derek L. Welton<br>C.R. Baugh, Ph.D. (Alt.) |
| Lockheed Martin Global Telecom | Mark T. Neibert<br>Prakash Chitre (Alt.) |
| Sprint – Long Distance Division | Thomas G. Croda<br>James Lord (Alt.) |
| WorldCom | Yi-Shang Shen<br>J. Martin Carroll (Alt.) |

**MANUFACTURERS**

| Organization Represented | Name of Representative |
|---|---|
| 3COM | Fred Lucas<br>Richard L. Stuart (Alt.) |
| ADC Telecommunications Inc. | Mike Rude |
| Airspan Communications Corp. | Douglas M. McCallister<br>Chris Rogers (Alt.) |
| Alcatel USA Inc. | Ken Biholar<br>Roz Sahakian (Alt.) |
| Aware, Inc. | Marcos Tzannes<br>William Meyer (Alt.) |
| Broadcom Corporation | David C. Jones<br>Aidan O'Rourke (Alt.) |
| Centillium Communications, Inc. | Dr. Syed Abbas<br>Guozhu Long (Alt.) |
| Cisco Systems, Inc. | John McDonough<br>Chip Sharpe (Alt.) |
| Conexant Systems, Inc. | Quentin C. Cassen |
| Copper Mountain Networks | Joseph D. Markee<br>John Reister (Alt.) |
| ECI Telecom Inc. | Ron Murphy<br>Todd Poole (Alt.) |
| Elastic Networks, Inc. | Patrick H. Stanley, P.E.<br>Jack Terry (Alt.) |
| Ericsson Inc. | Linda Troy<br>Stephen Hayes (Alt.) |
| Excelsus Technologies Inc. | Frederick Kiko<br>Don Robert House (Alt.) |
| Fujitsu America Inc. | Kenneth T. Coit<br>Hirohiko Yamamoto (Alt.) |
| General Datacomm Inc. | Fred Cronin<br>Mike McLoughlin (Alt.) |
| Globespan Semiconductor, Inc. | Massimo Sorbara<br>Clete Gardenhour (Alt.) |
| Harris Corp. | Marlis Humphrey<br>Tony Harb (Alt.) |
| Hekimian Laboratories | William H. Duncan |
| Hewlett-Packard | Karen Higginbottom |
| Hughes Network Systems, Inc. | Dr. Leonard Golding<br>Enrique Laborde (Alt.) |
| Lucent Technologies | Dave R. Andersen<br>Greg Ratta (Alt.) |
| Marconi Communications | Mark Scott<br>David K. Brown (Alt.) |
| Mayan Networks | Farooq Raza<br>Kevin W. Williams (Alt.) |
| Megaxess, Inc. | John Boal<br>Mihnea Nemes (Alt.) |

| Organization Represented | Name of Representative |
|---|---|
| Mitel Corp. | Silvana Rodrigues<br>Kelvin Steeden (Alt.) |
| Motorola Inc. | Syed Niaz<br>Dan Grossman (Alt.) |
| NEC America Inc. | Donovan Nak<br>Hajime Koto (Alt.) |
| Next Level Communications | Sabit Say<br>Jeffrey Weber (Alt.) |
| Nokia Telecommunications Inc. | Chris Wallace<br>Walt Tamminen (Alt.) |
| Nortel Networks | Mel N. Woinsky<br>Ed Eckert (Alt.) |
| OKI America Inc. | Henri Suyderhoud<br>Hisao Fujikawa (Alt.) |
| Paradyne Corp. | Richard K. Smith<br>Phil Kyees (Alt.) |
| PMC-Sierra, Inc. | Winston Mok<br>Terence Lau (Alt.) |
| Qualcomm Inc. | Mark Epstein<br>Ed Tiedemann (Alt.) |
| Siemens Information & Communications Networks, Inc. | David E. Francisco<br>Jim Stanco (Alt.) |
| ST Microelectronics | Jean-J Raynal<br>Roy Harvey (Alt.) |
| Symmetricom Inc. | Tony Pilarinos<br>Don Skipwith (Alt.) |
| Telecommunications Techniques | Michael Lewis<br>Jerry Gentile (Alt.) |
| Tellabs Operations, Inc. | Corey Parollina<br>Tom Rarick (Alt.) |
| Tellium, Inc. | Krishna Bala, PhD<br>Siegfried Giebl (Alt.) |
| Texas Instruments | James T. Carlo<br>Pete Chow, Ph.D. (Alt.) |
| TranSwitch Corp. | Jitender Vij<br>Edwin Soltysiak (Alt.) |
| Westell Technologies, Inc. | Guy Cerulli<br>Tariq Amjed (Alt.) |

At the time it approved this standard, Technical Subcommittee T1S1 on Services, Architectures & Signalling, which is responsible for the development of this standard, had the following members:

B. Hall, T1S1 Chair
G. Ratta, T1S1 Vice Chair

| Organization Represented | Name of Representative |
|---|---|
| ADC Telecommunications Inc. | Sal Morlando<br>Paul Krischlunas (Alt.) |
| Alcatel USA Inc. | Jeff Copley |
| AT&T | Doris S. Lebovits<br>John Keselica (Alt.) |
| AT&T Broadband | Sohan Grewal<br>Jim Dahl |
| Bell Atlantic | Dana Shillingburg<br>Michael Brusca (Alt.) |
| Bell Canada | Stewart Patch<br>P. Norman Smith (Alt.) |
| BellSouth Telecommunications Inc. | Robert V. Epley<br>David Whitney (Alt.) |
| CSI Telecommunications | Michael S. Newman<br>William J. Buckley (Alt.) |
| Cisco Systems | Dan Greene<br>Sue Geyer (Alt.) |
| Compaq Computer Corp. | John L. Schantz<br>Anantha Ramu (Alt.) |
| Defense Information Systems Agency | Don Choi<br>Ralph Liguori (Alt.) |
| Ericsson Incorporated | Linda Troy |
| Fujitsu America Inc. | Doug Hunt<br>Kenneth T. Coit (Alt.) |
| General Datacomm Inc. | Mike McLoughlin |
| GTE Telephone Operations | Michael Collison<br>John Rollins (Alt.) |
| Harris Corporation | Marlis Humphrey<br>Tony Harb (Alt.) |
| Hekimian Laboratories | William H. Duncan |
| Hewlett-Packard | James G. Baker |
| ICG Communications | Thomas Tardy<br>Kenneth Frederick (Alt.) |
| Illuminet | Kenn Moisey |
| Inet Technologies Inc. | Mart Nurmet<br>Said Saadeh (Alt.) |
| LG Sansys, Inc. | Hee Joung Lee<br>Mark Hosford (Alt.) |

| Organization Represented | Name of Representative |
|---|---|
| Lockheed Martin Global Telecom | Mark T. Neibert<br>Andy Gallant (Alt.) |
| Lucent Technologies | Robert B. Waller<br>Greg Ratta (Alt.) |
| Mayan Networks | Farooq Raza<br>Santu Muller (Alt.) |
| Megaxess, Inc. | John Boal<br>Mihnea Nemes (Alt.) |
| National Communications System | Nicholas Andre<br>Dale Barr (Alt.) |
| NEC America Incorporated | Kuei Y. Kou<br>Donovan Nak (Alt.) |
| Nokia Telecommunications Inc. | Jean-Luc Bouthhemy<br>Walt Tamminen (Alt.) |
| Nortel Networks | Mel N. Woinsky<br>Lewis C. Robart (Alt.) |
| OKI America Incorporated | Henri Suyderhoud<br>Hisao Fujikawa (Alt.) |
| Oresis Communications, Inc. | Michael R. Zeug<br>George Shenoda (Alt.) |
| Paradyne Corporation | Richard K. Smith<br>Phil Keyes (Alt.) |

| Organization Represented | Name of Representative |
|---|---|
| Qwest | Steve Showell<br>James L. Eitel |
| Rhythms | Rand Kennedy<br>David Reilly (Alt.) |
| SBC Communications, Inc. | B.S. Sambasivan<br>Clifton Campbell (Alt.) |
| Siemens Information and Communication Networks, Inc. | David LaMaster<br>Ron Franks (Alt.) |
| Sprint – Long Distance Division | James Lord<br>Albert D. Du Ree (Alt.) |
| Telcordia Technologies | Selvan Rengasami<br>Wesley Downum (Alt.) |
| Tellabs Operations, Inc. | Jim Orme<br>Mike Wurst (Alt.) |
| Tellium, Inc. | Krishna Bala, PhD<br>Siegfried Giebl (Alt.) |
| US Telecom Association (USTA) | Paul Johnson<br>Donald G. Bender (Alt.) |
| Voicestream Wireless Corp. | Albert H. Yuhan, Ph.D.<br>Gary K. Jones (Alt.) |
| WorldCom | Yatendra Pathak<br>Bernard Ku (Alt.) |

Sub Working Group T1S1.3 (Network Capabilities), which developed this standard, had the following active participants:

Wesley Downum, T1S1.3 Chair
Rich Hemmeter, T1S1.3 Network Capabilities Convener

Jeff Copley

Ranga Dendi

Stuart Goldman

William H. Krall

Ceyhan Lennon

Stewart Patch

Yatendra Pathak

Kraig Sanders

Dana Shillingburg

Ray P. Singh

Rajendra P. Udeshi

Scott Wilson

## Table of Contents

## Table of Figures

## Table of Tables

American National Standard
for Telecommunications –

# Signalling System Number 7 (SS7) – Upper Layer Security Capability

## 1 Scope, Purpose, and Application

The Security capability allows an end user service in the originating Signalling Point (SP) to invoke various security functions in the originating and/or destination SP. The Security capability can be used for identification and authentication of the communicating entities. It also provides information that supports resource access control, system access control, and encryption and decryption functions. The Security capability may be invoked by a variety of services. The end user will interact with an end user service which may invoke the Security capability. Note that the specific end user service that invokes Security is not within the scope of this capability description. The Security capability is not visible to the end user, but allows an end user service to take place. Thus, there is a "layering" of services and capabilities. The specification of the different security functions (e.g., identification, authentication, encryption) may be determined by the service or the Signalling point. The information exchange model and information element model for the Security network capability are based on the standards in the Reference section of this description.

A security policy is a statement of the rules that are to be enforced regarding the accessibility of data items and processing functions to entities within the network. In order to state the policy in a meaningful way, it is necessary to mention the threats that the policy is intended to prevent. Informative Annex B provides a threat analysis and security policy overview for Signalling System Number 7 (SS7).

Informative Annex C provides an overview of basic security functions and operation.

## 2 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

T1.114-2000*, Telecommunications - Signalling System No. 7 (SS7) - Transaction Capabilities.*[1]

ANSI X3.92-1981, *American National Standard Data Encryption Algorithm*.[2]

ANSI X9.9-1986, *Financial Institution Message Authentication*.[2]

CCITT Recommendation X.208-1988, *Specification of Abstract Syntax Notation One (ASN.1)*. [3]

[1] This document is available from the Alliance for Telecommunications Industry Solutions, 1200 G Street N.W., Suite 500, Washington, DC 20005. <http://www.atis.org>

[2] This document is available from the InterNational Committee for Information Technology Standards (INCITS). < http://www.techstreet.com/ncitsgate.html >

ISO/IEC 10181-2:1996, *Information technology -- Open Systems Interconnection -- Security frameworks for open systems: Authentication framework*.[4]

ISO/IEC IS 10745, *Information Technology - Open Systems Interconnection Upper Layers Security Model*, June 1993.[4]

ISO/IEC DIS 11586, *Information Technology - Generic Upper Layers Security*, June 1993.[4]

# 3 Definitions & Abbreviations

## 3.1 Definition of Terms

**3.1.1 Authorization Information**: Authorization is the process which allows an entity access to resources or services based on provided credentials. Authorization information contains an authorization value (e.g., login ID, password). It may also contain an authorization algorithm ID, indicating how to use the value (e.g., how to decrypt an encrypted value). Resource and System Access Control may base its decision to grant or deny the access request based on identification and authentication information.

**3.1.2 Confidentiality Information**: Confidentiality information is the information that is used in the destination signalling point to perform the necessary function(s) to prevent unauthorized disclosure of the component portion of a Transaction Capabilities Application Part (TCAP) message. It may contain a confidentiality algorithm ID, which identifies the appropriate confidentiality algorithm in the destination signalling point. It may also contain additional information needed to complete the specified confidentiality function (i.e., to allow encryption/decryption of the component portion of a TCAP message).

**3.1.3 Destination Signalling Point**: The Destination Signalling point contains the destination application. The Security network capability in the Destination Signalling point performs the received Security function indicated in the message from the Originating Signalling point.

**3.1.4 End User**: The End User is the subscriber to one or more services which invoke the Security capability.

**3.1.5 Identification and Authentication**: Identification and Authentication is a process whereby a claimed identity is verified by the Security capability. This process occurs when the end user, the service, or the Originating Signalling point provides a publicly known identifier and a private authenticator to the Security capability.

**3.1.6 Integrity information**: Integrity information is the information that is used in the destination signalling point to check whether or not the received message has been modified. It contains the Integrity Algorithm ID, which identifies the appropriate integrity algorithm in the destination signalling point. It may also contain additional information necessary to complete the specified integrity function.

**3.1.7 Originating Signalling Point**: The Originating signalling point contains the originating application process, which invokes a Security function in the destination signalling point.

[3] This document is available from the International Telecommunications Union.
< http://www.itu.int/ITU-T/ >

[4] This document is available from the International Organization for Standardization.
< http://www.iso.ch/iso/en/prods-services/ISOstore/store.html >

**3.1.8 Resource Access Control**: The Resource Access Control function in the destination signalling point may make use of Security information to grant or deny a request for access to resources or data at that signalling point. An application process which cannot be accessed except after passing the Resource Access Control function is "protected."

**3.1.9 Security Context**: The Security Context indicates the appropriate interpretation for the Confidentiality, Integrity, and the Authentication portion of Authorization information.

**3.1.10 Sequence Number**: The Sequence Number uniquely identifies a message in a sequence of messages. This information can be used to detect message insertions, deletions or replays in a transaction where many messages are exchanged.

**3.1.11 System Access Control**: System Access Control is the process of authorizing and continuing a transaction. System access control includes such features as timeout of idle transaction and transaction establishment based on the time of day or day of week.

**3.1.12 Time Stamp**: The Time Stamp value indicates when the message was sent by the originating signalling point. This information can be used, e.g., to detect message insertions, deletions, or replays in a transaction where many messages are exchanged.

### 3.2 Abbreviations & Acronyms

| | |
|---|---|
| ANSI | American National Standards Institute |
| ATIS | Alliance for Telecommunications Industry Solutions |
| CBC | Cipher Block Chaining |
| CCITT | International Telegraph and Telephone Consultative Committee |
| CCS | Common Channel Signalling |
| DEA | Data Encryption Algorithm |
| DES | Data Encryption Standard |
| D-H | Diffie-Hellman |
| FE | Functional Entity |
| GULS | Generic Upper Layer Security |
| ISO/IEC | International Organization for Standardization/International Electrotechnical Commission |
| ITU-T | International Telecommunication Union – Telecommunication Standardization Sector |
| MD2 | Message Digest 2 |
| MD5 | Message Digest 5 |
| RSA | The Public Key Cryptosystem invented by Rivest, Shamir, and Adleman |
| SCCP | Signalling Connection Control Part |
| SCP | Service Control Point |
| SDL | Specification and Description Language |

| | |
|---|---|
| SEP | Signalling End Point |
| SP | Signalling Point |
| SS7 | Signalling System Number 7 |
| TC | Transaction Capabilities |
| TCAP | Transaction Capabilities Application Part |

# 4 Description of Network Capability

## 4.1 General Description

The Security capability provides functions for identification and authentication of communicating entities. In addition, it provides resource access control, system access control, and capabilities to ensure secure transmission of information between the peer application processes (e.g., encryption and decryption). The end user will interact with an end user service which may invoke the Security capability. The specific end user service that invokes Security is not within the scope of this capability description.

## 4.2 Procedures

### 4.2.1 Provision/Withdrawal

From an end user's viewpoint, services requiring security to communicate with each other can use the Security capability. An end user cannot directly subscribe to the Security capability, but may subscribe to an end user service that uses the Security capability.

### 4.2.2 Normal Procedures

#### 4.2.2.1 Activation/Deactivation

Activation/Deactivation may be done on a service (instance)-by-service (instance) basis.

#### 4.2.2.2 Invocation and Operation

Invocation and Operation of the Security capability will be handled by an application process at the originating signalling point. An end user service determines the need for Security based on the application process being accessed in the destination signalling point. For example, access to an application process in the destination SP may be necessary to retrieve information necessary to execute the end user service. If the application process in the destination SP is protected (i.e., it allows access only to parties that pass the Resource Access Control function), the end user service in the originating SP must invoke the Security capability as a part of the information access process.

To invoke the Security capability, an application process must provide appropriate information to the Security network capability in the originating SP. An agreeable security context must be proposed before information related to other security functions can be processed. In addition, the application process may provide Authorization Value, Confidentiality, Integrity, Sequence Number, Time Stamp, or Key Exchange information, based on the negotiated security arrangements between the peer application processes. The Security capability at the originating SP can send the Security information (to invoke one or more Security functions at the destination SP) in conjunction with any necessary end user service-specific information to the destination SP.

The Security network capability at the destination SP receives the information and performs the requested security functions. If the functions are successful, the service-specific information is forwarded to the destination application process. The Security information is also available to the destination application process.

### 4.2.3 Exceptional Procedures

The Security capability may be invoked by an application process at the destination signalling point when the destination application process requires security functions not invoked by the originating application process.

The Security function(s) invoked may include functions performed at the originating node, the destination node, or both. If the destination SP cannot perform a requested Security function, or in the absence of a request for a required function, it will return an appropriate error or problem indication to the service at the originating SP which invoked Security.

A destination SP that is not equipped with the Security capability which receives a request to perform a Security function will use the normal protocol mechanisms for rejecting unrecognized information.

#### 4.2.3.1 Activation/Deactivation

None identified.

#### 4.2.3.2 Invocation and Operation

None identified.

### 4.2.4 Alternate Procedures

None identified.

#### 4.2.4.1 Activation/Deactivation

None identified.

#### 4.2.4.2 Invocation and Operation

None identified.

### 4.2.5 Interworking Considerations

None identified.

### 4.2.6 Network Capabilities for Charging

None identified.

### 4.2.7 Interactions with Supplementary Services

The Security capability should be transparent with respect to other services.

### 4.2.8 SDLs

The SDL diagram for the end user service is shown in Figure 1.

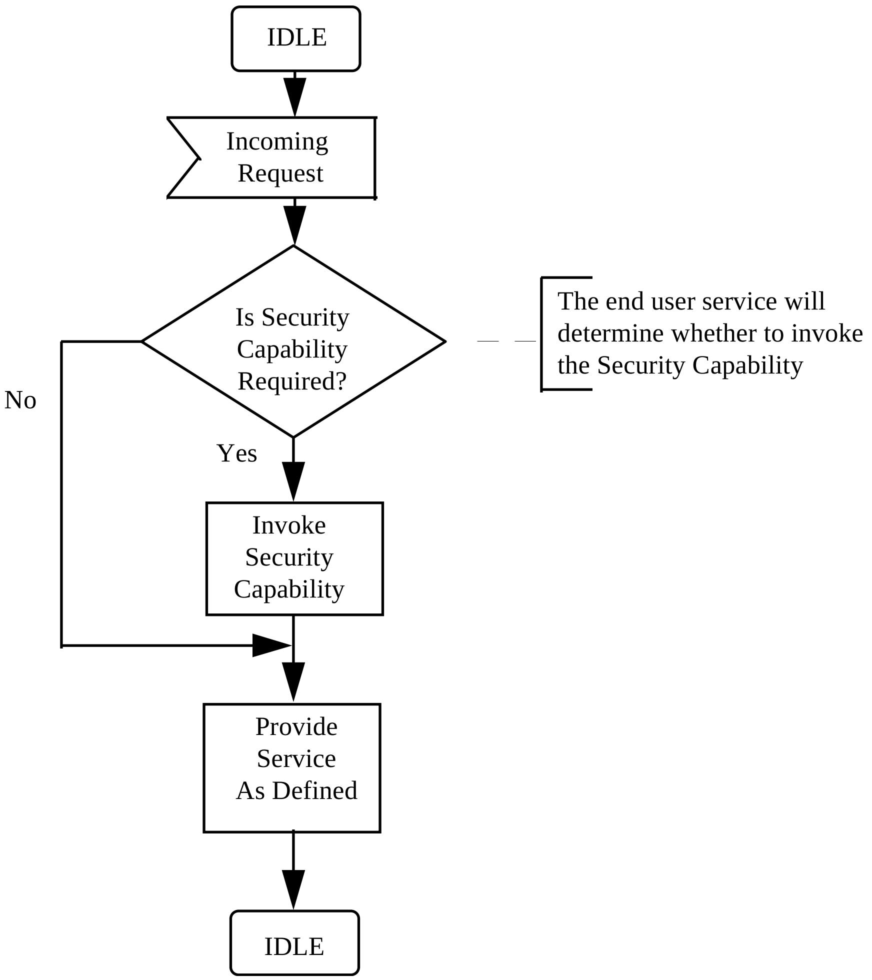
IDLE
Incoming Request
Is Security Capability Required?
The end user service will determine whether to invoke the Security Capability
No
Yes
Invoke Security Capability
Provide Service As Defined
IDLE

**Figure 1 - SDL Diagram for the End User Service**

# 5 Functional Capabilities and Information Flows

## 5.1 Functional Entity Model

The Security capability allows a functional entity (FE) to invoke various security functions in the originating and/or destination SPs. The functions supported by the Security capability include identification and authentication, resource access control, system access control, and encryption and decryption. Security functions required at the originating and destination functional entities are determined by the security strategy[5] negotiated by node management on behalf of the communicating application processes before the Security capability is invoked.

Some security functions (e.g., encryption) may be performed before sending information to a destination functional entity. Invocation of such security functions by the originating application process may require that the application process provide information that is not included in the outgoing information. The Security capability in the originating FE must also furnish security information to the destination FE so that proper security treatment can be applied to the received message. This security information must contain a security context. The originating application process may also send additional information to support resource access control, system access control, and encryption/decryption functions in the destination functional entity.

When the destination FE receives an invocation of the Security capability, the context information in the received security information determines the nature of the treatment and how Confidentiality, Integrity, and Authentication information is to be used. In addition, if the received security information contains confidentiality information, the confidentiality function specified by this information must be performed before other security information can be processed. The destination FE may contain additional data required to perform Security functions, e.g., an expected authorization value or a secret (symmetric) key for decryption.

If all invoked security functions in the destination FE complete without error (e.g., the identity of the sending entity is successfully validated), destination FE processes any non-security information from the incoming information flow.

### 5.1.1 Description of Originating Functional Entity

The originating FE is in an originating signalling point. The originating FE invokes one or more security functions in the destination functional entity. The originating application process is the source of the security information required to perform the invoked function(s). The Security capability requires that the originating application process provide security context information. The context information provides context for the security information. The originating application process may also provide identification/authentication, confidentiality, integrity, sequence number, time stamp, and key exchange information.

The authentication information is specific to the security context value. The confidentiality information contains a value identifying the confidentiality function and any additional information necessary to perform the specified confidentiality function within a specific security context. The integrity information

[5] The security strategy and the mechanisms associated with performing the necessary security functions are developed prior to the implementation of the Security capability in a node. The negotiation of the security strategy is not done in real-time.

contains a value identifying the integrity function and any additional information necessary to perform the specified integrity function within a specified security context. The sequence number uniquely identifies a message in a sequence of messages. The Time Stamp value indicates when the message was sent by the originating functional entity. Both the sequence number and the time stamp information can be used to detect message insertions, deletions or replays in a transaction where multiple messages are exchanged. The use of time stamp information to detect message insertion, message deletion or message replay will only work if the systems at both ends have secure “synchronized” clocks. Use of time stamp information to detect message insertions or deletions, is performed at the application layer. The key exchange information allows the sharing of information used to generate cryptographic keys.

### 5.1.2 Description of Destination Functional Entity

The destination FE is in a destination signalling point. The Security capability in the destination FE performs the security function(s) indicated by the received security information and those required by its security strategy.

## 5.2 Information Flow Model

To invoke the Security capability, an originating application process provides the appropriate security information. The information contains the data necessary to invoke the necessary security functions both at the originating FE and at the destination functional entity. It also contains the Security Context information used to establish a proper security context. If no security context is proposed, or if the proposed security context is not acceptable, the destination FE cannot perform the security functions requested by the originating functional entity. Once established, a security context applies for the duration of the interaction between the originating FE and the destination functional entity, unless a new security context is agreed.

Once the proper security treatment (if required) has been applied at the originating functional entity, the security information, in addition to non-security related information, is sent to the destination functional entity.

The information flow diagram is shown in Figure 2.

When the destination FE receives the security information, it performs the invoked security functions within the agreed security context. This may include returning security information to the originating FE (e.g., to provide an acknowledgment or a prompt for further information.) If all invoked security functions in the destination FE complete without error (e.g., the identity of the sending entity is successfully validated), the message contents are sent to the appropriate application process.

If an error is encountered, the destination FE sends an appropriate security error back to the originating functional entity.

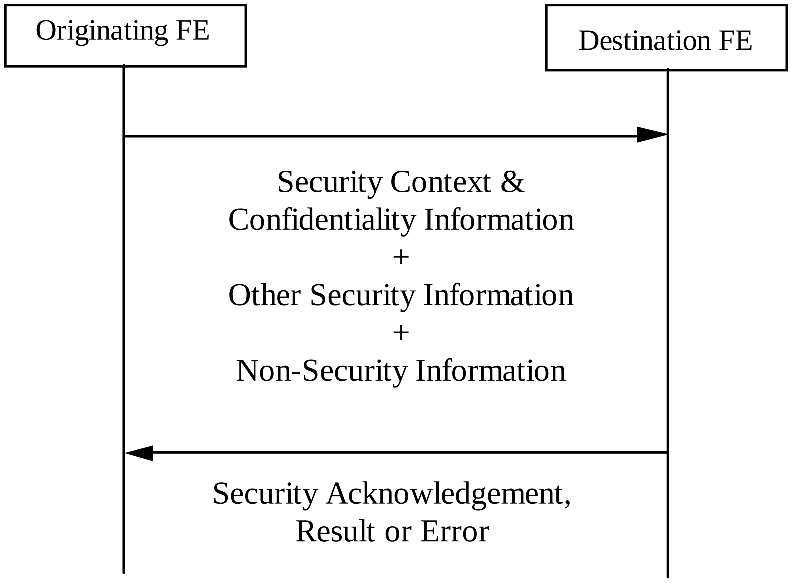

**Figure 2 - Security Information Flow Diagram**

### 5.2.1 Invoking Security in the Originating Functional Entity

The originating FE invokes the security functions (based on the security strategies) to ensure that proper security treatment is applied before the information flow is sent to the destination functional entity, (e.g., the originating application process may invoke an encryption function).

### 5.2.2 Invoking Security in the Destination Functional Entity

Security information that is received at the destination FE invokes the specified security functions. The originating application process can invoke security functions at the destination FE by including such information in a message. The destination application process may have access to some security functions locally (e.g., to fail a message that does not contain required security information).

### 5.2.3 Activation and Deactivation of Security

Activation and deactivation of Security is done on a FE basis. If Security is invoked in a FE where it is not active, the SS7 message invoking Security will be discarded and an appropriate error message returned to the sender. If an information flow that does not invoke Security is received at a FE where Security is active, the destination FE may invoke one or more security functions before acting on the information.

### 5.2.4 Exceptional Procedures

If the destination FE cannot successfully perform a requested security function, or if the request for a required function does not include an invocation of a required security function, the destination functional entity will return an appropriate error or problem code to the service at the originating FE that invoked the Security.

The ability to negotiate security context, algorithms, or parameters, or to perform key exchange is not supported in this standard. To provide such capabilities, one may refer to international standards (e.g., ITU-T Recommendation X.509. Information Technology - Open Systems Interconnection- The Directory: Authentication Framework. 1993.) In addition, informative annexes C, E, and F provide specific examples of how this negotiation may take place.

### 5.2.5 Allocation of Functions to Equipment

The security functions may be at any Signalling End Point (SEP).

## 6 Protocol and Procedures

Informative Annex D contains an example of the use of this protocol and these procedures for secret key cryptography.

Informative Annex E contains an example mechanism for exchanging secret key information.

Informative Annex F contains an example of the use of this protocol and these procedures for public key cryptography.

### 6.1 Protocol and Procedural Assumptions

– Since the Security capability does not involve any message routing, it should be a part of the Transaction Capabilities (TC) user protocol rather than Signalling Connection Control Part (SCCP).

– Since, for this network capability, message decryption must be performed before the components of a message can be read, security context and Confidentiality information should be part of TC, not the TC user.

– Since some services will not invoke the security capability, Security should not impose additional protocol or procedure when it is not invoked.

– Security may be invoked in conjunction with other operations.

– TC-user-level security mechanisms may also be appropriate if, for example, only a portion of a component is encrypted. This is beyond the scope of this standard.

### 6.2 Format of the Security Information

### 6.2.1 Format of the Security Information in the Dialogue Portion

Security information is optional in the dialogue portion and takes the following form:

| | |
|---|---|
| Security Context Parameter | O |
| Confidentiality Parameter | O |

#### 6.2.1.1 Format of the Security Context Parameter

| |
|---|
| Security Context Parameter Identifier |
| Security Context Parameter Length |
| Security Context Parameter Value |

The Security Context provides the context for Security. The context information determines how other security information should be interpreted. Once an application process has sent a security context, that context applies to all subsequent messages[6] sent by either application process during the transaction, until one of the application process sends a message indicating a different security context.

The Security Context may be an integer or an object identifier type. Private security contexts are object identifiers. Standard security contexts may be integers or object identifiers. The standard integer values for security context are defined as follows:

**Table 1 - Integer Security Context Values**

| Value | Meaning |
|---|---|
| 0 | Not Used |
| 1 | Data Encryption Algorithm (DEA) (Note 1) |
| 2 | Reserved for Integrity Check |
| 3 | Message Code Authentication (Note 2) |
| 4 | DEA and Message Code Authentication (Notes 1 and 2) |
| 5 | Sequence, Timestamp and/or Identification Only (Note 3) |
| > 5 | Reserved |

NOTE 1 - ANSI X3.92-1981, *American National Standard Data Encryption Algorithm*. The determination of the DEA mode, kind of padding and whether an Initialization Vector or confounder is used are for further study.

NOTE 2 - ANSI X9.9-1986, *Financial Institution Message Authentication*.

NOTE 3 - Sequence, Timestamp or Identification may be present with the other integer security contexts. This context is used when these parameters are the only ones used.

[6] Note that security information in an Abort message is ignored and any Security context will not apply to an Abort message since the Abort has no components.

### 6.2.1.2 Format of Confidentiality Parameter

| |
|---|
| Confidentiality Parameter Identifier |
| Confidentiality Parameter Length |
| Confidentiality  AlgID Identifier |
| Confidentiality AlgID Length |
| Confidentiality AlgID |
| Confidentiality Value Identifier |
| Confidentiality Value Length |
| Confidentiality Value |

The Confidentiality parameter contains the information within a specific context that should be used in the destination SP to perform the necessary confidentiality function(s). The Confidentiality Algorithm ID identifies the algorithm that is to be used in the destination SP to decipher data that has been encrypted by the originator. The Confidentiality Value parameter contains additional information (i.e., Confidentiality Algorithm Value) necessary to complete the specified confidentiality function.

Once an application process has sent a confidentiality parameter or has otherwise established mutually-acceptable confidentiality information with its peer process, the selected confidentiality algorithm and confidentiality value apply to all subsequent messages (except Abort) sent by either application process during the transaction, until one of the application process sends a message indicating different values. When this occurs, the new algorithm or value replaces the previous value (i.e., the new algorithm is not applied in addition to the previous algorithm).

The confidentiality function specified by the confidentiality parameter applies to all components in a TC package. Thus, if the Security information in a TC package indicates encryption, the components of the message (i.e., everything after the Component Sequence Length) will be encrypted. The network capability does not support the selection of certain components for encryption while other components in the same TC package are unencrypted or encrypted differently.

The Confidentiality Value consists of one or more octet strings and is optional within the Confidentiality parameter.  The Confidentiality Algorithm ID is an integer that ranges from 0 to 127 or is an Object Identifier.  The ID is optional within the Confidentiality parameter.  If the Confidentiality Algorithm ID is not included in the Confidentiality parameter, a default algorithm (which may be determined based on the Security Context) should be used.  The Confidentiality parameter is not sent with empty contents.

### 6.2.2 Format of the Security Operation

| |
|---|
| Security Operation Identifier |
| Security Operation Length |
| Parameter Sequence Identifier |
| Parameter Sequence Length |
| Authorization Parameter |
| Integrity Parameter |
| Sequence Number Parameter |

| |
|---|
| TimeStamp Parameter |
| KeyExchange Parameter |

NOTE - This table does not include the component type and component identifier(s) included as part of the TCAP message.

The security information is transferred as parameters in the Security Operation. The Security Authorization Value, Integrity, Sequence Number, Time Stamp, and KeyExchange parameters are optional. However, at least one parameter must exist in a Security Operation. A Security Operation without parameters shall not be sent. For example, if Confidentiality information is the only Security-related information in a message, no Security operation shall be included.

### 6.2.2.1 Format of Authorization Parameter

| |
|---|
| Authorization Parameter Identifier |
| Authorization Parameter Length |
| Authorization AlgID Identifier |
| Authorization AlgID Length |
| Authorization AlgID |
| Authorization Value Identifier |
| Authorization Value Length |
| Authorization Value |

The Authorization parameter contains initiator-provided access control information in the form of an Authorization Algorithm ID and an Authorization Value. The Authorization Algorithm ID identifies the algorithm that is to be used in the destination SP to authenticate the sending entity. For example, if the Authorization Value is an encrypted certificate, the Authorization Algorithm ID would include the necessary decryption information. Integer values of the Authorization Algorithm ID are reserved for algorithms identified in this ANSI standard. Object Identifier values are available for other use.

The Authorization Value contains the identification and authentication information (e.g., login, password). The Authorization Value is determined by the application process invoking the Security capability. The value can be any unique information valid in the two communicating application processes, such as a service value or user ID. The duration of an agreed identification or authentication is controlled by the application process granting the identification or authentication. The Authorization Algorithm ID is an integer that ranges from 0 to 127 or is an Object Identifier. The ID is optional within the Authorization parameter. If the Authorization Algorithm ID is not included in the Authorization parameter, a previously agreed algorithm should be used. The default algorithm may be determined based on the Security Context. The Authorization Value is structured as specified by the Authorization Algorithm and is mandatory within the Authorization parameter.

Examples of the use of authorization information may be found in T1 TR.40-1995, *Security Requirements for Electronic Bonding between Two TMNs*[1], clause 4.2.

### 6.2.2.2 Format of Integrity Parameter

| |
|---|
| Integrity Parameter Identifier |
| Integrity Parameter  Length |
| Integrity AlgID Identifier |
| Integrity AlgID Length |
| Integrity AlgID |
| Integrity Value Identifier |
| Integrity Value Length |
| Integrity Value |

The Integrity parameter contains the information within a specific context that should be used in the destination SP to check whether or not the received message has been modified. It contains the Integrity Algorithm ID, which identifies the appropriate integrity algorithm in the destination SP. It may also contain additional information (e.g., checksum) necessary to complete the specified integrity function.

The structure and content of the integrity value is specified as part of the definition of an integrity algorithm.   The optional Integrity Algorithm ID identifies the mechanism to be applied to the Integrity value and is an integer that ranges from 0 to 127 or is an Object Identifier.  If the Integrity Algorithm ID is not included in the Integrity parameter, a default algorithm (which may be determined based on the Security Context) should be used.

### 6.2.2.3 Format of Sequence Number Parameter

| |
|---|
| Sequence Number Parameter Identifier |
| Sequence Number Parameter Length |
| Sequence Number Parameter Value |

The Sequence Number uniquely identifies a message in a sequence of messages. This information may be used to detect message insertions, deletions or replays in a transaction where many messages are exchanged.

The Sequence Number Value is an integer.

### 6.2.2.4 Format of Time Stamp Parameter

| |
|---|
| TimeStamp Parameter Identifier |
| TimeStamp Parameter Length |
| TimeStamp Parameter Value |

The Time Stamp value indicates when the message was sent by the originating SP. This information may be used to detect message insertions, deletions, or replays in a transaction where many messages are exchanged.

The Time Stamp parameter is defined in T1.114.5-2000, clause 4.1.

#### 6.2.2.5 Format of KeyExchange Parameter

| |
|---|
| KeyExchange Parameter Identifier |
| KeyExchange Parameter Length |
| KeyExchange AlgID Identifier |
| KeyExchange AlgID Length |
| KeyExchange AlgID |
| KeyExchange Value Identifier |
| KeyExchange Value Length |
| KeyExchange Value |

The KeyExchange parameter contains information related to the process of key exchange. It specifies the algorithm being used and any values that need to be passed. The KeyExchange Algorithm ID identifies the mechanism being used to exchange a key. It is either an Object Identifier or an integer ranging from 0 to 127. Committee T1 will assign the values 0 through 127 to key exchange algorithms as needed. If an Algorithm ID is not included in the KeyExchange parameter, a default algorithm should be used. The KeyExchange Value consists of one or more octet strings to be used as, or in the generation of, a cryptographic key.

### 6.3 Procedures for Security

#### 6.3.1 Actions at the Originating Signalling Point

The application process at the originating signalling point invokes the Security capability. Based on the security strategies of the originating and the destination SPs, the originating application process indicates to the originating SP the security treatment, if any, required prior to sending the Security Operation. In addition, it will also provide context and confidentiality information and the parameter values for the Security Operation parameters. The mechanism by which the originating application process determines the security treatment at the originating SP or the security information that is sent to the destination SP are beyond the scope of this network capability description.

If encryption is required, the components of the message are encrypted.

Once the proper security treatment has been performed, the information associated with sending the Security Operation is passed to TC.

#### 6.3.1.1 Actions for Integrity at the Originating Signalling Point

If Integrity procedures are desired, the originating SP populates the Integrity Algorithm ID field with an identifier indicating what algorithm is being used. The node populates the Integrity Value field with the result obtained by applying the selected integrity algorithm on the entire component portion with the exception of the Integrity Parameter and the Security Operation Length field.

#### 6.3.1.2 Actions for Key Exchange at the Originating Signalling Point

If cryptographic key exchange is desired, the originating signalling Point populates the KeyExchange Algorithm ID field with an identifier indicating what key exchange algorithm is desired (e.g., Diffie-Hellman). The KeyExchange Value field is populated with the first value (or values) required by the particular algorithm. For instance, Diffie-Hellman (D-H) requires two integers to be passed. Alternatively if a public key scheme is being used, the originating node may populate the field with its public key.

### 6.3.2 Actions at an Intermediate Signalling Point

None identified.

### 6.3.3 Actions at the Destination Signalling Point

The Security Context parameter provides the context for the Security Operation and the Confidentiality Parameter. The context information indicates how the security information is to be interpreted. If the received message includes a confidentiality parameter or if currently active context implies encryption, the contents of the TC component portion are decrypted.

The received Security Operation invokes the Security capability. The information contained in the Security Operation is used to perform the various security functions. The optional Authorization Value can contain data needed to perform identification and authentication functions. If included in the Security Operation, the other optional parameters -- Integrity, Sequence Number, and Time Stamp -- are used to indicate specific security functions and values associated with the functions that are required to verify the sending entity. If all security functions in the destination node complete without error (e.g., the identity of the sending entity is successfully validated), the non-security information of the message is made available to the receiving application process.

#### 6.3.3.1 Actions for Integrity at the Destination Signalling Point

If Integrity procedures are being used, the destination signalling Point extracts the Integrity Algorithm ID and, if required, the Integrity Value. It then uses the indicated algorithm to detect any modification of the message. Typically this entails applying the selected algorithm on the message and comparing the result to the received Integrity Value.

#### 6.3.3.2 Actions for Key Exchange at the Destination CCS Node

Upon receipt of the KeyExchange parameter, the destination node extracts the KeyExchange Value and the KeyExchange Algorithm ID. Based on the Algorithm ID (and the corresponding key exchange mechanism) the destination manipulates the KeyExchange Value. For instance, if the exchange is using

D-H, the destination node performs some computations using the KeyExchange Value and responds to the originating node (using the KeyExchange parameter) with the appropriate result.

If a node is unable to recognize the KeyExchange parameter it is expected that the operation will fail.

### 6.3.4 Error Conditions

If Security Context is not supported, the transaction will be aborted. Depending on the version of TC, the mechanism for this will be a Reject component (T1.114-1988), a P-Abort cause of “Unrecognized Dialogue or Component Portion Identifier” (T1.114-1992[7]) or “Security context not supported” (T1.114-1996 and beyond).

If Confidentiality Algorithm is not supported or the Confidentiality Value is not valid with the specified Algorithm, the transaction shall be aborted with a U-Abort cause of "abnormal security."

If the requested security operation cannot successfully be performed, an appropriate error or problem code shall be returned in a Return Error or Reject Component.

If the particular proposed Security Context is not supported, the transaction shall be aborted with a U-Abort cause of “Security Context not supported.”

If a protocol error is detected in the Security component, the component shall be Rejected.

[7] This error was called “Unrecognized Component Portion Identifier” in T1.114-1992.

**Annex A**

(normative)

# A Security Information Definition in ASN.1

## A.1 Format of the Security Information

### A.1.1 Format of Security Information in the Dialogue Portion.

```
DialoguePortion ::= [25] IMPLICIT SEQUENCE {
                              ProtocolVersion           OPTIONAL,
                              ApplicationContext        OPTIONAL,
                              SecurityContext           OPTIONAL,
                              ConfidentialityOPTIONAL
                                                        }
```

### A.1.1.1 Format of Security Context Parameter

```
SecurityContext ::= CHOICE {
        [0] IMPLICIT INTEGER
        [1] IMPLICIT OBJECT IDENTIFIER}
```

| Integer Value | Meaning |
|---|---|
| 0 | Not Used |
| 1 | Data Encryption Algorithm (DEA) (Note 1) |
| 2 | Reserved for Integrity Check |
| 3 | Message Code Authentication (Note 2) |
| 4 | DEA and Message Code Authentication (Notes 1 and 2) |
| 5 | Sequence, Timestamp and/or Identification Only (Note 3) |
| > 5 | Reserved |

NOTE 1 - ANSI X3.92-1981, *American National Standard Data Encryption Algorithm*. The determination of the DEA mode, kind of padding and whether an Initialization Vector or confounder is used are for further study.

NOTE 2 - ANSI X9.9-1986, *Financial Institution Message Authentication*.

NOTE 3 - Sequence, Time Stamp or Identification may be present with the other integer security contexts. This context is used when these parameters are the only ones used.

### A.1.1.2 Format of Confidentiality Parameter

```
Confidentiality ::= [2] IMPLICIT SEQUENCE{
        ConfidentialityAlgID OPTIONAL,
        ConfidentialityValue OPTIONAL,
        }



ConfidentialityAlgID ::= CHOICE{
        [0] IMPLICIT INTEGER (0..127),
        [1] IMPLICIT OBJECT IDENTIFIER }

ConfidentialityValue ::= ANY DEFINED BY ConfidentialityAlgID
```

### A.1.1.3 User Abort Information

```
UserAbortInformation ::= [24] IMPLICIT SEQUENCE {
UserAbortCause
}

UserAbortCause ::= IMPLICIT ENUMERATED {
          abnormalDialogue    (0),
          abnormalSecurity    (1),
          securityCapabilityNotSupported (2),
          securityContextNotSupported (3),
           ...
```

### A.1.2 Format of the Security Operation

```
          security              OPERATION
          PARAMETER             SEQUENCE{
                                Authorization           OPTIONAL,
                                Integrity         OPTIONAL,
                                SequenceNumber          OPTIONAL,
                                TimeStamp               OPTIONAL,
                                KeyExchange             OPTIONAL
                                                        }
          ERROR{unexpectedComponentSequence,
                                unavailableResource,
                                missingCustomerRecord
                                taskRefused
                                securityError
                                missingParameter
                                unexpectedParameterSequence
                                unexpectedMessage
                                unexpectedPackageType}
                                ::= 8 3
```

### A.1.2.1 Format of Authorization Value Parameter

```
          Authorization ::= [29] IMPLICIT SEQUENCE{
                                AuthorizationAlgID OPTIONAL,
                                [2]  IMPLICIT AuthorizationValue,
                                }
```

```
AuthorizationAlgID ::= CHOICE{
                    [0] IMPLICIT INTEGER (0...127),
                    [1] IMPLICIT OBJECT IDENTIFIER
                    }

AuthorizationValue ::= ANY DEFINED BY AuthorizationAlgID
```

### A.1.2.2 Format of Integrity Parameter

```
Integrity ::= [30] IMPLICIT SEQUENCE{
                     IntegrityAlgID OPTIONAL,
                    [2] IntegrityValue,
                    }

IntegrityAlgID ::= CHOICE{
          [0] IMPLICIT INTEGER (0..127),
          [1] IMPLICIT OBJECT IDENTIFIER
          }


IntegrityValue ::= ANY DEFINED BY IntegrityAlgID
```

### A.1.2.3 Format of Sequence Number Parameter

```
SequenceNumber ::= [31] IMPLICIT INTEGER
```

### A.1.2.4 Format of Time Stamp Parameter

```
TimeStamp ::= UTCTime   -- CCITT Recommendation X.208-1988 defines UTCTime.
```

### A.1.2.5 Format of KeyExchange Parameter

```
KeyExchange ::= [32] IMPLICIT SEQUENCE{
                    KeyExchangeAlgID OPTIONAL,
                    [2] KeyExchangeValue,
                    }

KeyExchangeAlgID ::= CHOICE{
          [0] IMPLICIT INTEGER (0..127),
          [1] IMPLICIT OBJECT IDENTIFIER
          }


KeyExchangeValue ::= ANY DEFINED BY KeyExchangeAlgID
```

### A.1.2.6 Error Codes

```
unexpectedComponentSequence                  ERROR ::= 1
                 unexpectedDataValue         ERROR ::= 2
                 unavailableResource         ERROR ::= 3
                 missingCustomerRecord       ERROR ::= 4
                 taskRefused                 ERROR ::= 7
                securityError               ERROR ::= 18
                missingParameter            ERROR ::= 19
                unexpectedParameterSequence ERROR ::= 20
                unexpectedMessage           ERROR ::= 21
               unexpectedPackageType       ERROR ::=  22
```

## Annex B
(informative)

# B Threat Analysis and Security Policy Overview

## B.1 Introduction

A security policy is a statement of the rules that are to be enforced regarding the accessibility of data items and processing functions to entities within the network. In order to state the policy in a meaningful way, it is necessary to mention the threats that the policy is intended to prevent.

Two classes of security threats are identified: threats to the underlying network, and threats to the applications, or services, that are built on top of the network. This standard describes a framework within which defenses against both classes of threats can be implemented. In addition, it describes a set of basic security functions that defend against the first class -- threats to the underlying network.

The following discussion of security policy and threat analysis is present here to assist users of the standard in understanding which threats are addressed by the standard, and by implication, which threats are left to be addressed by application designers and implementers. It should be noted that the discussion below does not imply that all of the identified security problems will be solved by the security provisions in this standard. Rather, this discussion attempts to describe the security threats in breadth prior to stating which ones are and are not addressed by the standard.

## B.2 Threat and Policy Categories

There are four major threat categories: (1) theft of information; (2) unauthorized alteration of information; (3) sabotage of the network (denial of service); and (4) theft of service. The four corresponding areas of security policy are: (1) confidentiality; (2) integrity; (3) availability; and (4) usage authorization. In a network, attention should be given separately to the threats to end-user information and network control information, in each of the four threat categories. Security features that support the four major areas of security policy include user authentication, message integrity, message confidentiality, and access control. A lower layer of supporting security mechanisms includes such things as encryption, message digests, digital signatures, and secure operating systems.

### B.2.1 Methods of Attack

Attacks on the network can be carried out by three methods: physical access to network nodes or links, network access to network nodes, and the introduction of malicious software during the software development or software distribution processes.

In addition, individual applications can be attacked at the end user interface. Attempts can be made to exploit weaknesses in the user authentication and usage authorization features of the application. Such threats must be defended by application-specific security measures. They are not addressed by the basic security functions specified in this standard, nor are they discussed any further in this threat analysis.

### B.2.2 Physical Attacks

Attacks based on physical access to nodes could be carried out by insiders abusing their authorized access to nodes, by employees abusing their building access to gain unauthorized access to nodes, or by intruders who breach building security. Having gained access to a node, an intruder or insider could alter hardware or software, or make use of maintenance interfaces. It is possible to steal end-user or network control information, alter both types of information, or sabotage the node. Having access to -- and unlimited control over -- a node, an intruder can use it to launch network-based attacks on other nodes.

### B.2.3 Network-Based Attacks

Network-based attacks could come from a compromised node in a telephone company's own SS7 network. With the advent of mediated access, they could also come from the networks of other telephone companies or third party service providers, due either to unscrupulous insiders or to lax physical security that allows intruders to gain access to nodes. In addition, an intruder having physical access to a link could attach computing equipment to it, and use that equipment to carry out network-based attacks.

There are two categories of network-based attacks: passive and active. Passive attacks involve the monitoring of messages, and the theft of end-user or network control information. Active attacks involve the sending of messages, often with forged sender IDs. These messages are calculated to induce the receiver to take some improper action, that will result in a successful attack in one of the four threat areas (theft or alteration of information, denial of service, or theft of service). Often such messages exploit known bugs in the software in the receiving node.

### B.2.4 Software Development/Distribution Attacks

Unscrupulous software developers will sometimes insert Trojan horses or trapdoors into their programs. These are pieces of malicious code that will carry out some covert function when they are installed in a production system, possibly including allowing the author to break into the system, bypassing its security features.

## B.3 Consequences of Attacks

Successful attacks in any of the threat areas could allow the perpetrator to accomplish any of the following:

- Theft of private end-user information, such as voice conversations, voice mail, or data.

- Theft of private telephone company information, such as customer lists, calling card numbers, or cellular authentication codes.

- Alteration of end-user or telephone company information for the purpose of damaging the information resources of the victim.

- Theft or alteration of, network control information to facilitate further penetration of the network.

- Selective interference with the services of certain individuals or firms, for purposes of harassment or unscrupulous competition.

- Widespread interference with network services (i.e., sabotage), or the threat of it, for purposes of terrorism or extortion.

- Theft of telephone services.

## B.4 Security Policies

A variety of security policies could be implemented to defend against the above attacks. The choice of policies is, to some extent, a business decision involving cost-benefit considerations. Different applications will in general have different security requirements. It is noticed, however, that message confidentiality and message integrity are two basic security functions needed by most applications. This standard addresses the security mechanisms used to provide confidentiality and integrity. Security mechanisms to provide the ability to negotiate security context, algorithms, or parameters, or to perform key exchange are outside the scope of this standard. Annex C describes a set of basic security functions and makes recommendations for their use.

## Annex C
(informative)

# C Overview of Basic Security Functions and Operation

## C.1 Introduction

The basic security functions include message confidentiality and message integrity. These functions were chosen to make up the basic set because they are commonly needed by most applications, and because their implementation does not involve any application-specific considerations.

These functions defend against the physical attacks and network-based attacks described in Annex B. These attacks are threats to all applications. Message confidentiality prevents eavesdroppers on the SS7 network from stealing sensitive information. Message integrity prevents the successful alteration or forging of messages.

Other security functions (such as end user authentication) are dependent on application-specific information, so it is difficult to standardize their details. These functions are not included in the basic set. Their design is left to the designers of the applications that need them.

The confidentiality and integrity functions both depend on cryptography. In the case of secret key (symmetric) security mechanisms, each pair of communicating entities (e.g., SS7 network nodes) possess a secret key (e.g., a DES key) that is known to both entities in the pair and is not known to any other entity.

To reduce the risk that keys could be compromised either by theft or by cryptanalysis, keys must be changed periodically. The problem of getting keys into the hands of authorized entities while keeping them out of the hands of all other entities is the key management problem. The chosen key management protocol does not involve a central security server, and is intended to be carried out periodically (perhaps every day or every few days) between nodes that have a need to communicate with each other. The key management transaction is not intended to be associated with any particular application transaction. Instead, it is carried out asynchronously, preferably at a time of light load. The resulting key can be used to provide security for any application transactions between that pair of nodes.

If both nodes are capable of generating and validating digital signatures, then the Authenticated D-H protocol will be used. If either node does not support digital signatures, then the D-H protocol will be used. (The latter is somewhat less secure.) Thus, prior to key establishment, there will be a negotiation to decide which protocol is to be used. There are other security functions that may or may not be supported by both nodes. The initial negotiation will determine these as well.

The term "security association" has been chosen to describe the mutually agreed set of security functions and the shared keys between two nodes. Security association establishment consists of two parts: the negotiation to agree on a set of mutually supported security functions, and the establishment of shared secret keys. Periodically, the second part will be repeated to establish new keys; this is called "key renewal."

Using the shared keys, message confidentiality is provided using -- for example -- the DES algorithm, and message integrity is provided using DES combined with one of the standard message hash algorithms. The following sections describe these functions in detail.

## C.2. Security Association Establishment and Key Management

Before secure communication can take place between a pair of nodes, a security association must be established between them. A security association consists of two parts: (1) an agreement on a set of security functions that are supported by both nodes; and (2) an exchange of information resulting in the establishment of a cryptographic key (or keys) known only to that pair of nodes.

From time to time, the key establishment portion -- or the entire security association establishment -- can be repeated, for a variety of reasons that will be discussed below. Key establishment involves a significant amount of computation, and should therefore be done infrequently -- daily, or less often. Because association establishment, including key establishment, will be time-consuming, a transaction supporting services to a paying customer should not be held up awaiting security association establishment.

### C.2.1 Overview of Security Association Establishment

A node can send a request for security association establishment to any other node, at any time. The request will include a proposed list of security functions to be supported. A positive response will include a list of acceptable security functions. This list must be equal to, or a subset of, the set proposed in the initial request. After agreeing on functions, the two nodes will exchange messages containing key establishment information for those security functions that the two nodes have in common.

When two nodes have an established security association, either one can send a request for a renewal of the established keys. This request will be identical to the first key establishment message in the above sequence. A positive response will be identical to the last message in the above sequence.

A negative response can be given to either request. Reasons can include: (1) this node does not yet support any security functions; or (2) this node is currently too busy to do a key establishment computation, and, for a time, either (a) wishes to continue using the current keys, or (b) wishes to communicate without security.

This standard does not specify an action to be taken when a request receives a negative response. The intention is not to refuse communication or deny service to customers when a node is unable to support security functions. Rather, it is to provide a way for nodes that do support security functions to interoperate in order to communicate with each other securely.

A decision to refuse insecure communication would be based on management policies outside the scope of this standard. During the transition period, when security functions are being introduced into the network and many nodes do not yet support them, the default should clearly be to accept insecure communication.

### C.2.2 Occasions for Security Association Establishment

When a node is first added to the network, or when it is brought back online after an outage, it will have no security associations with any other nodes. There are several alternatives for establishing new security associations. The choice between them should be based on performance criteria, and is not specified by this standard. The first is for the new node to immediately attempt to establish a security association with every other node in the network. The second is for the new node to begin establishing security associations with a selected list of nodes in order to be prepared to communicate securely with them. The list would be provided during the installation or initialization process. The third is to wait for an application on the new node to request communication with other nodes, and establish security associations only as needed. In any case, as noted earlier, serving customers should take precedence over establishing security associations.

To reduce the risk that encryption keys could be compromised, either by theft or by cryptanalysis, the keys should be changed at suitable intervals. The determination of "suitable" depends on two things: (1) the ability of the involved nodes to support the processing load of key establishment; and (2) the estimated seriousness of the key theft or cryptanalysis threat.  A suitable interval is not specified by this standard. It is suggested that it be no more often than daily, and no less often than several days. It is also suggested that steps be taken to randomize the times of key changes. If every node in the network were to request a key change of every other node at the same time (e.g., at 2:00 AM), the resulting computational load would have a serious impact on performance, even at a time of very light customer load.

Some nodes might lack sufficient memory to store security association information for every node they have ever communicated with. Such nodes would have to re-use memory, discarding information on the least recently used association in order to establish a new association. Thus, it may occur that one node in a pair retains memory of their association while the other one has "forgotten" it. Such an occurrence will be discovered either when the forgetful node requests establishment of a new association, or the other node sends a secure message that the forgetful node is unable to decrypt. Both nodes must be prepared to recognize this case and re-establish their association, and temporarily use insecure communication in order to avoid delaying service to customers.

After the basic security functions have been implemented in all network nodes, and legacy equipment has been upgraded as required, it might be practical for each node to maintain a security association with every other node, and to reject all insecure communication as probably being from an intruder. This would be most desirable from a security point of view.

### C.2.3 Examples of Security Association

Informative Annex D contains an example of secret (symmetric) key cryptography.

Informative Annex E contains an example mechanism for exchanging private key information.

Informative Annex F contains an example of the public key cryptography.

## C.3 Message confidentiality and Integrity

An application process may request message confidentiality, message integrity, or both when sending a message to any destination. If a security association has been established with that destination, then the requested functions will be provided.

In this example, message confidentiality will be achieved by encrypting the message, using DES in CBC mode, and using the key shared by the sending and receiving nodes.

In this example, message integrity will be achieved in one of three ways, depending on the negotiation of supported functions. If MD5 is supported by both nodes, then an MD5 digest of the message will be computed, encrypted using DES with the shared key, and sent to the destination along with the cleartext message. Otherwise, if MD2 is supported by both nodes, then the above will be done, with MD2 replacing MD5. Otherwise, DES will be run over the message in CBC mode, and the CBC residue (the last block) will be sent along with the cleartext message.

If both confidentiality and integrity are requested, then there are only two alternatives. DES-CBC alone cannot provide both confidentiality and integrity  (assuming a single shared symmetric key). A message digest (either MD5 or MD2) will be computed and concatenated with the message, and both encrypted, using DES-CBC, with the shared key.

If no security association has been established with the destination, then obviously the requested security functions cannot be provided. The action to be taken in this case is not specified by this standard. Suggested actions include sending the message without security and initiating a security association establishment with the destination node so that future messages may be exchanged securely. Notification of the application process would also seem desirable, but that, again, is a matter not specified by this standard. (It is suggested that, in notifying the application process, this case be treated as a non-fatal error.)

### C.4 Summary of Advisory Items

This annex contains recommendations for when and how the basic security functions should be used. The use of security functions by an application is optional. Some applications will have a strong need for security, some will have little need for it, and some will be unable to operate while using it.

#### C.4.1 Recommendations for Basic Security Services

The basic security functions -- message confidentiality and message integrity -- should be implemented in all SS7 nodes so that they will be available to those applications that need them.

During the transition period, when security functions are being introduced into the network, not all nodes will support them. Therefore, applications running on nodes that do support security should be prepared to continue operating normally when communicating with nodes that do not yet support security.

If an application transmits sensitive information over the network, then that application should use message confidentiality. Sensitive information is information that could cause harm if it were to fall into the wrong hands. Two examples of sensitive information are given in the following paragraphs.

When a customer uses a calling card to pay for a long distance call, the calling card number, and the customer's PIN, are transmitted over the network. If those numbers were to fall into the hands of an unauthorized person, they could be used to commit toll fraud. Incidents of toll fraud involving theft of card numbers off the network have been reported in the press in recent years. Card numbers and PINs should be protected by message confidentiality.

The problem of cellular fraud by means of cloning is well known. Plans are underway to combat this problem by securing the cellular air link and authenticating cellular phones cryptographically. When a cellular customer roams outside his home area, the authentication information must be sent from his home location, over the network, to the area where he is roaming. Once the air link is secured, so that authentication information can no longer be stolen off the air, attempts will probably be made to steal authentication information off the network. This information should be protected by message confidentiality.

Message integrity prevents the alteration or forging of messages. Two classes of messages can be suggested as requiring integrity: (1) messages that authorize the use of services; and (2) messages that change a customer's service parameters (such as the number to which the customer's calls should be forwarded).

#### C.4.2 Recommendations for Security Association Establishment

A node can send a request for security association establishment to any other node, at any time. That request can be accepted or rejected. It is mandated by the standard that the requester shall be prepared for the request to be rejected.

Reasons for rejection could include: (1) security functions are not implemented; (2) too busy now; (3) no room in database for more security associations; or (4) other reasons, not yet determined.

Because of legacy equipment of low memory and processing capacity, and because of inevitable delays in implementing new functionality throughout the network, it is anticipated that rejections of security association establishment requests will be frequent, if they are issued indiscriminately. It is therefore suggested that carriers make use of the basic security functions on an ad hoc basis to attempt to bring some of the more serious security problems under control, while operating within the constraints imposed by legacy equipment. The following paragraph gives an example of such an ad hoc solution, to illustrate what is meant by this suggestion.

It is suspected that most nodes will have neither the processing capacity nor the memory capacity to establish and maintain a security association with every other node in the network. However, it might not be necessary to do that in order to solve actual security problems. Consider the examples above, in which calling card numbers and PINs were identified as needing confidentiality, and messages authorizing the usage of services were identified as needing integrity. Suppose there were a single SCP containing a large database of calling card numbers, along with other information necessary for making an authorization decision. Perhaps that SCP has the necessary processing and memory capacity to maintain a security association with every switch in the network -- or perhaps that one SCP could be upgraded to have the necessary capacity. Given this set of security associations, the messages from the switches to the SCP, which contain calling card numbers and PINs, could be protected with confidentiality. Similarly, the messages from the SCP to the switches, giving authorization for calls to be completed, could be protected with integrity. (A similar ad hoc solution can be envisioned to protect cellular authentication information by establishing security associations between cellular switches.) It is anticipated that such ad hoc solutions will evolve toward more extensive and formal key and credential management infrastructures as warranted, such as the use of certificates and a central certification authority as described in international standards. (See, for example, ITU-T Recommendation X.509, *Information Technology - Open Systems Interconnection- The Directory: Authentication Framework*, 1993.)

Annex D

(informative)

# D Symmetric Key Encryption/Decryption

## D.1 Introduction

The Security capability allows an end user service in the originating signalling node to invoke various security functions in the originating and/or destination signalling nodes. This annex describes a symmetric key encryption implementation of the Security capability.

Symmetric key cryptography makes use of a single shared key. Messages encrypted using a symmetric key are computationally expensive to decrypt without the symmetric key. The scheme is based on the assumption of a shared secret. It is assumed that only certain entities possess knowledge of the symmetric key. Therefore, successfully decrypting a message using the shared symmetric key identifies the sender. A sender desiring confidential communication between himself and another encrypts his message with the key. Many symmetric key algorithms also support integrity through the use of “digital signatures.”

> NOTE - The term “digital signatures” is more commonly used in the context of asymmetric key cryptography. The term is used in this annex for convenience.

Some representation of the message (e.g., a checksum or a hash) is encrypted using symmetric key. The result is the signature and is passed along to the destination. At the destination the key is used to decrypt the signature. The result is compared to the message to detect modification.

In the sections which follow, an example of how symmetric key cryptography might be used for confidentiality and integrity are presented.

## D.2 References

Schneier, Bruce. *Applied Cryptography*. New York, NY: John Wiley & Sons, 1996, Second Edition.

## D.3 Confidentiality

This section demonstrates how the security capability could be used to provide confidentiality using symmetric key encryption. This example uses DES in Cipher Block Chaining mode (DES-CBC) as its data encryption algorithm.

Prior to the time when secure communications are desired, the destination and originating signalling points need to share a symmetric key. A symmetric key needs to be shared by every pair of nodes with which secure communications are desired.

When confidential communications are desired, the originating signalling point encrypts the component portion using the using symmetric algorithm (i.e., DES-CBC), and then invokes the security capability. The security information in the dialog portion has the following form:

The Security Context Parameter reflects that symmetric key cryptography is being used by having the Security Context Parameter Value set to a value of "6."

The presence of the Confidentiality Parameter indicates that confidentiality is being invoked. The algorithm is identified in the Confidentiality AlgID parameter as DES in CBC mode. Additionally, the Confidentiality Value uses the Point Code parameter and the Initialization Vector Parameter to identify the originator of the message and to provide the initialization vector.

| |
|---|
| Confidentiality Parameter Identifier |
| Confidentiality Parameter Length |
| Confidentiality AlgID Identifier (optional) |
| Confidentiality AlgID Length (optional) |
| Confidentiality AlgID (if not DES-CBC) |
| Confidentiality Value Identifier |
| Confidentiality Value Length |
| Point Code Parameter Identifier |
| Point Code Parameter Length |
| Point Code = Originating Point Code |
| Initialization Vector Parameter Identifier |
| Initialization Vector Length |
| Initialization Vector Value |

The Confidentiality parameter contains the information within a specific context that should be used in the destination signalling node to perform the necessary confidentiality function. For Security Context 6, the default Confidentiality Algorithm is DES in CBC. The Confidentiality Value is a constructor containing the two elements required to decrypt the component portion of the SS7 message. The Point Code of the originator is used at the destination node to determine which symmetric key is to be used to decrypt the message. The Initialization Vector is needed to begin decryption.

Upon receipt of each encrypted message the receiving node uses Point Code of the sender to select the appropriate symmetric key. It uses the key and the Initialization Vector to decrypt the component portion of the message.

## D.4 Integrity

Just as with Confidentiality, prior to the time when secure communications are desired, the destination and originating signalling points need to share a symmetric key.

When communications with assured integrity are desired, the originating signalling point calculates a signature and invokes the security capability. In this example the signature is constructed by a method indicated in the Integrity AlgID. In the first method, the component portion (except for the Integrity Parameter and the Security Operation Length field) is hashed using a one-way hashing algorithm -- in this case MD2 or MD5. In the second case, the signature is constructed using the CBC residue. The result is encrypted using the shared symmetric key. The final result is the signature.

The presence of the Integrity Parameter indicates that integrity is being invoked. The Integrity parameter contains the information within a specific context that should be used in the destination signalling node to check whether or not the received message has been modified. The Integrity Algorithm ID identifies the integrity algorithm and the Integrity Value is populated with the digital signature.

Within Security Context 6, the Integrity Algorithm ID takes the following values:

| Integrity AlgID | |
|---|---|
| 0000 0001 | MD5 |
| 0000 0010 | CBC residue |
| 0000 0011 | MD2 |

Upon receipt of the message, the destination node decrypts the signature (in the case of MD2 or MD5) using the shared symmetric key and compares it to the component portion. If they agree, integrity is assumed. If they do not, integrity should not be assumed.

Note that the use of signatures based on a one-way hashing algorithm (such as MD2 or MD5) requires an Initialization Vector. For signatures based on CBC residue, the Initialization Vector is set to 0 (not used).

## Annex E
(informative)

# E A Mechanism for Secret Key Exchange

## E.1 Introduction

Secret key cryptography requires a common key at the two ends of a transaction. This annex describes one mechanism for secure exchange of secret keys, using SS7. The Secret Key Exchange operation is defined below.

## E.2 References

ITU-T Recommendation X.509, *Information Technology - Open Systems Interconnection- The Directory: Authentication Framework*. 1993.[3]

## E.3 Secret Key Exchange

The Secret Key Exchange operation carries the indication of which key exchange protocol will be used during the key exchange, associated signing information, and any required certificate. It also carries an indication of how the secret key will be used by the Basic Security Functions that are described in Annexes C and D.

The Secret Key Exchange operation is part of the Procedural Family of operations and is encoded as national constructor:

| | |
|---|---|
| 1001 0000 | procedural family, reply required |
| 0000 0100 | Key Establishment |
| | length |
| | protocol and certificate indicator |
| | Diffie-Hellman information |
| | Certificate |

The protocol and certificate indicator is encoded as an octet string of length one as follows:

| H | G | F | E | D | C | B | A |
|---|---|---|---|---|---|---|---|
| operation code | | authentication protocol | | key exchange protocol | certificate needed | D-H information indicator | certificate indicator |

The following are the meanings of values in the protocol and certificate indicator:

Operation code (two bits):

00 Initial key establishment

| | |
|---|---|
| 01 | Response to initial key establishment |
| 10 | Key renewal |
| 11 | Response to key renewal |

Authentication protocol (two bits):

| | |
|---|---|
| 00 | Neither MD5 nor MD2 supported |
| 01 | MD5 only supported |
| 10 | MD2 only supported |
| 11 | Both MD5 and MD2 supported |

Key exchange protocol (one bit):

| | |
|---|---|
| 0 | Diffie-Hellman |
| 1 | Authenticated Diffie-Hellman |

Certificate needed (one bit):

| | |
|---|---|
| 0 | Sender already has a copy of receiver’s certificate |
| 1 | Sender needs a copy of receiver’s certificate |

Diffie-Hellman information indicator (one bit):

| | |
|---|---|
| 0 | D-H information not included in message |
| 1 | D-H information is included in message |

Certificate indicator (one bit):

| | |
|---|---|
| 0 | Certificate not included in message |
| 1 | Sender’s certificate is included in message |

The D-H information is encoded application-wide, primitive, and contains the signed (if authenticated D-H is in use) or unsigned (if D-H is in use) value of

$$(G^S) \bmod p$$

that is used by each end of the conversation to compute the secret key for use in a later conversation.

The certificate is encoded application-wide, constructor, and contains an X.509 certificate containing the sender’s identity and public key. Encoding of certificates derived explicitly from the sender’s SS7 node identity are for further study.

The Secret Key Exchange operation returns success or failure. On success, a Return Result is returned. The Return Result includes the same parameters as the initial message, with different values. The operation code will be 1 or 3, indicating response to initial key establishment or to key renewal. The authentication protocol field will (in response to initial key establishment) indicate the protocol chosen by the responder, and likewise for the key exchange protocol field. The certificate needed field will indicate the need of the responder, while the D-H information and Certificate indicator fields indicate the presence or absence of the respective items in the Return Result. Further detail on the content of the Return Result will be found below, under Procedures.

Errors that the Secret Key Exchange operation returns are:

| | |
|---|---|
| 0000 0011 | Unavailable Resource |

| | |
|---|---|
| 0000 0111 | Task Refused |
| 0000 1100 | Unauthorized Request |
| 0001 0011 | Missing Parameter |

## E.4 Procedures

### E.4.1 Procedures for Initial Key Establishment

For initial key establishment, four messages are sent between the initiator and responder—two in each direction.

To begin the exchange, the initiator sends the Secret Key Exchange operation, including the operation code (with a value of 00), the authentication protocol indicator (with a value indicating which protocols are supported by the initiator), the key exchange protocol (indicating whether or not authenticated D-H is supported by the initiator), and, if authenticated D-H is supported and the initiator does not already have a copy of the responder’s certificate, a value of 1 for the certificate needed indicator. The D-H information and certificate are not present in the initial message, so the D-H indicator and Certificate indicator are both = 0.

The responder sends a Return Result containing the same fields as above, with values as follows: operation code=01; authentication protocol equal to the best protocol supported by both initiator and responder (01 (MD5) if supported by both, else 10 (MD2) if supported by both, else 00, implying use of DES-CBC residue); key exchange protocol equal to the best protocol supported by both (1, indicating Authenticated D-H, if supported by both, else 0, implying unauthenticated D-H); certificate needed=1 if authenticated D-H is supported by both and responder needs initiator’s certificate. As in the initial message, the D-H information and certificate are not included, and so the respective indicators are = 0.

On receipt of the Return Result, the initiator sends the Secret Key Exchange operation again, with fields and values as follows: operation code=0; authentication protocol equal to that returned by the responder; key exchange protocol equal to that returned by the responder; certificate needed field as in first initiator message; D-H information indicator=1; certificate indicator as appropriate. The D-H information field will contain either signed or unsigned D-H information, as agreed. The certificate field will contain the initiator’s certificate if it was requested by the responder.

The responder sends a Return Result containing the same fields as above, with values as above except: operation code=1; Certificate indicator as requested by initiator. The D-H information field will contain the responderís D-H information. The certificate field will contain the responder’s certificate if it was requested.

### E.4.1 Procedures for Key Renewal

For key renewal, only two messages are sent—one in each direction—since the determination of mutually supported protocols took place at the time of initial key establishment.

To begin the exchange, the initiator sends the Secret Key Exchange operation, with fields and values as follows: operation code=10; authentication protocol and key exchange protocol equal to those previously agreed; certificate needed field equal to 0 (certificates having been previously exchanged if needed); D-H information indicator=1; certificate indicator=0. The D-H information field will contain either signed or unsigned D-H information, as agreed. The certificate field will not be present.

The responder sends a Return Result containing the same fields as above, with values as above except: operation code=11. The D-H information field will contain the responder’s D-H information.

**Annex F**
(informative)

# F Public Key Encryption/Decryption

## F.1 Introduction

The Security capability allows an end user service in the originating signalling node to invoke various security functions in the originating and/or destination signalling nodes. This annex describes a public key encryption implementation of the Security capability.

Public key cryptography makes use of two keys: one public and one private. Messages encrypted using a public key are computationally expensive to decrypt without the private key. A sender desiring confidential communication between himself and another encrypts his message with the other's public key. Many public key algorithms also support integrity through the use of digital signatures. Some representation of the message (e.g., a checksum or a hash) is encrypted using the sender's private key. The result is the signature and is passed along to the destination. The sender's public key is used to decrypt the signature. The sender's public key decrypts only messages encrypted with her private key. Hence, successful decryption using her public key serves to indicate that the signature did indeed come from her. The result is compared with the message checksum or hash value to detect modification.

Public key cryptography can be more computationally expensive than symmetric cryptography. In instances where this is a concern, public key cryptography provides a means by which symmetric keys can be distributed securely. The theft of a symmetric key compromises information for only as long as the key is being used. Public key cryptography provides a secure means to distribute symmetric keys, allowing for their more frequent change.

In the sections which follow, examples of how public key cryptography might be used for confidentiality (one using public key and one using a hybrid public and symmetric approach) and integrity are presented.

## F.2 References

Schneier, Bruce. *Applied Cryptography*. New York, NY: John Wiley & Sons, 1996, Second Edition.

Schneier, Bruce. *E-Mail Security, How to Keep Your Electronic Messages Private*. New York, NY: John Wiley & Sons, 1995.

Amoroso, Edward. *Fundamentals of Computer Security Technology*. Englewood Cliffs, NJ: PTR Prentice Hall, 1994.

## F.3 Confidentiality

This section demonstrates how the security capability could be used to provide confidentiality using public key encryption. The first example uses public key encryption to protect the data in the component portion. The second example demonstrates how public key encryption can be used in conjunction with symmetric key encryption. The data is encrypted using the symmetric algorithm and the symmetric key is passed using public key cryptography.

### F.3.1 Public Key

Prior to the time when secure communications are desired, the destination and originating signalling points each obtain a public and private key pair. Public keys need to be exchanged (e.g., by exchanging X.509 certificates) with every node with which secure communications are desired.

When confidential communications are desired, the originating signalling point encrypts the component portion using the destination SP's public key and then invokes the security capability.

In this example the Security Context Parameter Value is an object identifier indicating the public key algorithm being used (e.g., RSA).

The Confidentiality parameter contains the information within a specific context that should be used in the destination signalling node to perform the necessary confidentiality function. In this case the Confidentiality Algorithm ID might further specify the algorithm that is to be used in the destination signalling node to decipher data that has been encrypted by the originator.

### F.3.2 Public Key for Key Management

As in the previous section, prior to the time when secure communications are desired, the destination and originating signalling points each obtain a public and private key pair. Public keys need to be exchanged (e.g., by exchanging X.509 certificates) with every node with which secure communications are desired.

This time, when confidential communications are desired, the originating signalling point obtains a random symmetric encryption key and encrypts the component portion using a symmetric encryption algorithm. It then encrypts the random key using the destination SP's public key. It then invokes the security capability.

The Security Context Parameter Value is an object identifier indicating the scheme being used.

The Confidentiality parameter contains the information within a specific context that should be used in the destination signalling node to perform the necessary confidentiality function. In this case the Confidentiality Algorithm ID identifies the symmetric encryption algorithm to be used in the destination signalling node to decipher the data. The Confidentiality Value is populated with the public key encrypted symmetric key. When the symmetric key is extracted and decrypted it can then be used to decrypt the rest of the message using the symmetric algorithm. If an initialization vector is needed it can be passed as the Confidentiality Value.

Upon receipt of each encrypted message the receiving node uses its private key to decrypt the Confidentiality Value. It then uses the result to decrypt the component portion.

## F.4 Integrity

Just as with Confidentiality, prior to the time when secure communications are desired, the destination and originating signalling points each obtain a public and private key pair. Public keys are exchanged (e.g., by exchanging X.509 certificates) with each node with which secure communications are desired.

When communications with assured integrity are desired, the originating signalling point calculates the signature and invokes the security capability. In this example the signature is constructed by hashing the component portion (except for Integrity Parameter and the Security Operation Length field) using a one-way hashing algorithm, e.g. MD5. The resulting hash is encrypted using the originating node's private key. The final result is called a signature.

In this example, the Authorization Parameter, Sequence Number, and Time Stamp parameters of the Security operation are not required. However, a Time Stamp or Sequence Number parameter could be used to detect message replay.

The Integrity parameter contains the information within a specific context that should be used in the destination signalling node to check whether or not the received message has been modified. The Integrity Algorithm ID identifies the integrity algorithm. The Integrity Value is populated with the digital signature.

Upon receipt of the message, the destination node decrypts the signature using the originating node's public key, hashes the component portion and compares the two. If they match the integrity is assumed. If they do not, integrity should not be assumed.

## Annex G
(informative)

## G Bibliography

T1 TR.40-1995, *Security Requirements for Electronic Bonding between Two TMNs.*[1]

ITU-T Recommendation X.509, *Information Technology - Open Systems Interconnection- The Directory: Authentication Framework*. 1993.[3]