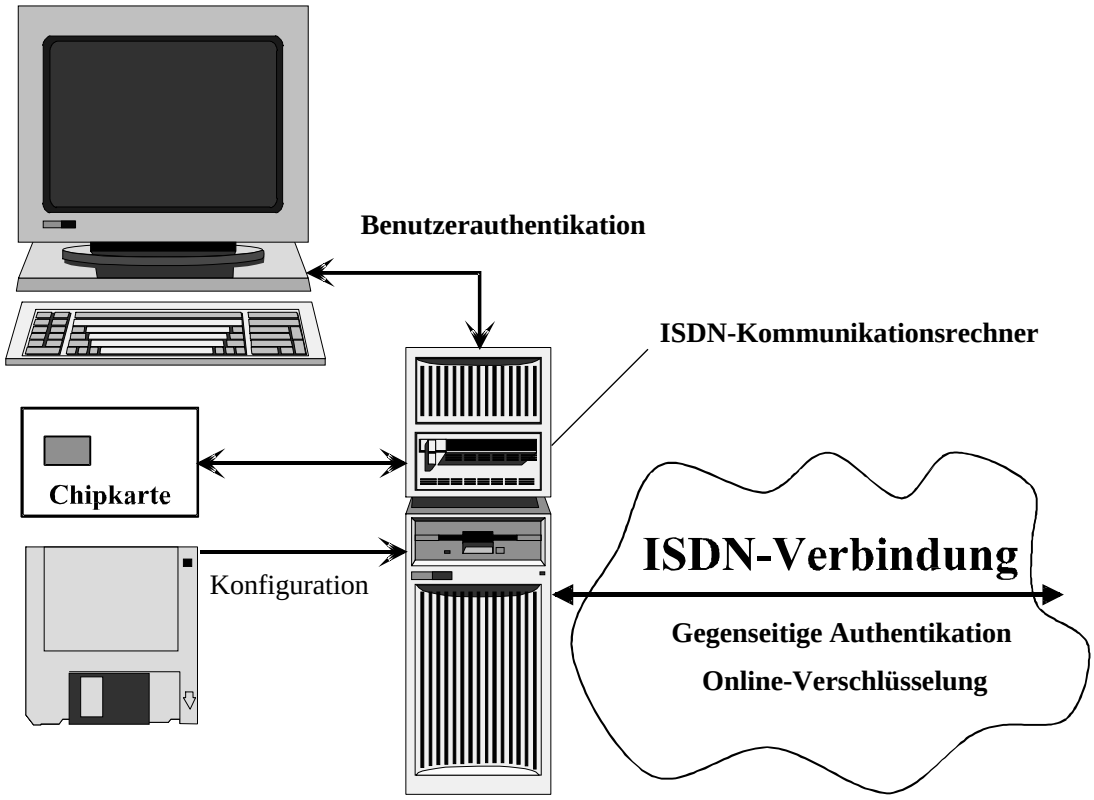
In: Kleinke, G. (Hrsg.): *Nutzung und Technik von Kommunikationsendgeräten*. ITG-Fachbericht 131, vde-verlag, Offenbach 1994, S. 81-95.

Transparente Sicherheitsmechanismen für ISDN-Anwendungen

Dienst, Detlef; Fox, Dirk; Ruland, Christoph
Institut für Nachrichtenübermittlung, Universität Siegen, D-57068 Siegen



Zusammenfassung

Mit wachsender Bedeutung der Datenübertragung in privaten und öffentlichen Netzen und dem Trend zu immer kleineren Kommunikationsendgeräten als "täglichen Begleitern" (Notebooks, PDAs) stellen sich Fragen nach der Sicherheit der angebotenen Dienste und Protokolle mit besonderer Dringlichkeit. Das Projekt S-CAPI verfolgt das Ziel, rechnerbasierte ISDN-Kommunikationslösungen mit kryptographisch starken Sicherheitsmechanismen auszustatten. Zu diesem Zweck wurden die Sicherheitsdienste Datenintegrität, Authentikation, Zugriffskontrolle und Vertraulichkeit für eine transparente Integration in rechnerbasierte ISDN-Anwendungen spezifiziert. Sie setzen auf dem *Common ISDN API* (CAPI) auf, das sich inzwischen zu einem Herstellerstandard bei der Entwicklung rechnerbasierter ISDN-Kommunikationslösungen durchgesetzt hat. Für DOS-Workstations wurde ein S-CAPI-Prototyp entwickelt.

Schlagworte:

ISDN, CAPI, Sicherheit, Vertraulichkeit, Authentifikation, Integrität.

1 Einleitung

Die Datenübertragung in öffentlichen Netzen spielt vor allem beim Einsatz moderner Kommunikationstechnik in größeren und mittleren Unternehmen eine immer wichtigere Rolle. Interne und externe Unternehmenskommunikation findet in wachsendem Umfang auf elektronischem Wege statt. Darunter fallen sowohl der Zusammenschluß lokaler Netzwerke über WAN-Verbindungen als auch die Integration von Kommunikationsanwendungen in bestehende IT-Systeme (Fax, *electronic mail*, Datex-P, Datex-J etc.). Dieser Trend erhält derzeit durch die zunehmende Verfügbarkeit von ISDN-Anschlüssen einen weiteren Schub.

In einer solchen kommunikationstechnischen Umgebung bekommen Fragen nach dem **Schutz der übertragenen Daten große Bedeutung**: So werden immer häufiger sensible Daten (Verträge, Verhandlungen, Geschäftskontakte) elektronisch übertragen, während zugleich die Kontrolle über die verwendeten Übertragungsmedien abnimmt (öffentliche Leitungen, Vermittlungsstellen, Richtfunk-, Mobilfunk- und Satellitenkanäle).

Das diensteintegrierende **digitale öffentliche Netz ISDN** nimmt bei dieser Entwicklung eine Schlüsselrolle ein, da es eine **sehr schnelle Übertragung beliebiger digitaler Daten** (64 kBit/s pro Kanal) über bestehende Teilnehmeranschlußleitungen ermöglicht. Die Übertragungskosten liegen dabei oft deutlich unter denen einer Modem-, Datex-P- oder Standleitungsverbindung. ISDN-Kommunikationsdienste (wie z.B. Fax, Filetransfer), auf *personal computers* (PCs), PDAs und *workstations* transparent zugänglich gemacht, werden zunehmend nachgefragt.

Ziel des **Projektes S-CAPI** ist es, **Sicherheitsdienste transparent in rechnerbasierte ISDN-Kommunikationslösungen zu integrieren**. Um sowohl von der eingesetzten ISDN-Hardware als auch von der Anwendung unabhängig zu bleiben, setzen die spezifizierten Sicherheitsmechanismen auf dem *Common ISDN Application Programming Interface* (CAPI) auf, einer Schnittstelle, die von deutschen ISDN-Hardware-Herstellern spezifiziert wurde.

An die zu spezifizierenden und implementierenden Sicherheitsdienste wurden die folgenden Anforderungen gestellt:

- Die **Integration der Mechanismen** sollte **vollständig transparent** erfolgen, d.h. unabhängig sowohl von der auf der CAPI aufsetzenden Kommunikationsanwendung als auch von der eingesetzten Hardware. Dabei sollte die **CAPI-Spezifikation nicht erweitert** werden, damit die Nutzung der S-CAPI-Sicherheitsdienste weder eine Modifikation der Anwendung noch des herstellerspezifischen CAPI-Schnittstellentreibers erfordert.
- Die **Sicherheitsmechanismen sollten existierenden Standards genügen** und ausschließlich **bewährte kryptographische Verfahren verwenden**. Statt proprietärer Algorithmen wurden daher eine Anzahl genormter und gut untersuchter Kryptosysteme implementiert und eingebunden.
- Die Mechanismen sollten auch für einen Einsatz in Lösungen **für Primär-Multiplex-Anschlüsse geeignet** sein, d.h. einen hohen Durchsatz ermöglichen (30 B-Kanäle mit zusammen 1920 kBit/s). Um dieser Randbedingung zu genügen, war die **Integration kryptographischer Hardware** vorzusehen und großes Gewicht auf die Verwendung effizienter Software-Implementierungen zu legen.

2 Das *Common ISDN Application Programming Interface*

Im September 1990 wurde die Version 1.1 des unter der Schirmherrschaft der Bundespost Telekom von verschiedenen deutschen ISDN-Hardware-Herstellern spezifizierten *Common ISDN Application Programming Interface* (CAPI) veröffentlicht. Diese Schnittstelle ermöglichte erstmalig eine hardwareunabhängige Implementierung von rechnerbasierten ISDN-Kommunikationsanwendungen. Sie konnte sich in den darauffolgenden Jahren in Deutschland als Herstellerstandard für die Entwicklung von ISDN-Anwendungen unter unterschiedlichen Betriebssystemen durchsetzen [1] und verschaffte deutschen ISDN-Entwicklungen erhebliche Marktvorteile.

Der CAPI-Standard besteht im Kern aus zwei Teilen: einer betriebssystemunabhängigen Spezifikation von Dienstprimitiven und einer Beschreibung der speziellen Kommunikationsschnittstellen für ausgewählte Betriebssysteme (Unix, DOS, Windows, OS/2, NetWare), die von ISDN-Hardware-Herstellern durch entsprechende Treiber unterstützt werden.

Zunächst ausschließlich für das nationale ISDN-Protokoll spezifiziert, wurde das CAPI nach Verabschiedung des ISDN-Protokoll-Standards der ETSI um die auf ETS 300 102 / Q.931 basierenden Protokolle und das DSS1-Protokoll ergänzt. Die überarbeitete Version 2.0 der Spezifikation liegt seit Februar 1994 vor [2]. Sie wurde inzwischen bei der *International Telecommunications Union* (ITU) als Standardisierungsvorschlag eingereicht [3]. Nicht zuletzt wegen der Mitwirkung von Novell, Alcatel, Microsoft und IBM an der neuen Spezifikation wird sich das CAPI voraussichtlich gegen alternative Vorschläge wie den kürzlich verabschiedeten ETSI-Standard *Programming Communication Interface* (PCI) international durchsetzen.

2.1 Das CAPI im OSI-Referenzmodell

Das CAPI bietet durch weitgehende Abstraktion von Eigenschaften der ISDN-Hardware und von Signalisierungs- und Protokollfunktionen einen einfachen und vereinheitlichten Zugriff auf ISDN-Dienste.

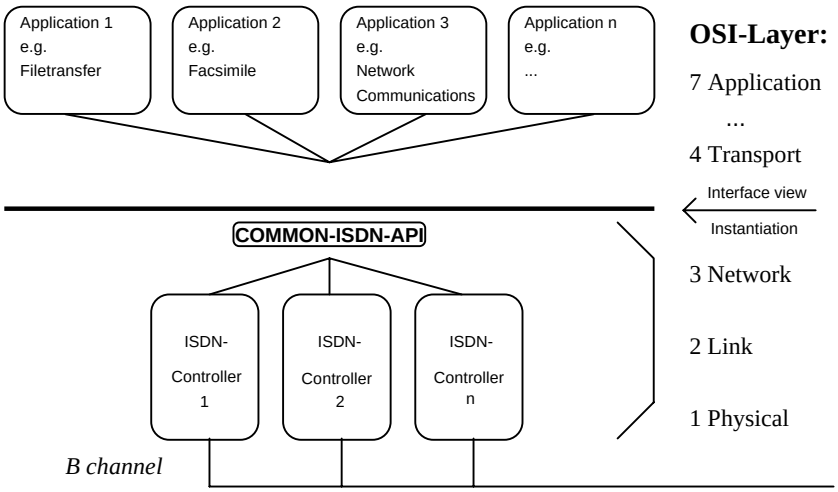


Bild 2-1: Einordnung des CAPI im OSI-Referenzmodell [2].

Aus der Sicht des OSI-Referenzmodells [4] liegt das CAPI oberhalb des *network layers*, d.h. es bietet darüberliegenden ISDN-Anwendungen transparente Ende-zu-Ende-Verbindungen auf einem B-Kanal (**Bild 2-1**). Dabei werden für die Datenübertragung - abhängig vom gewählten Dienst - unterschiedliche Kommunikationsprotokolle für die Schichten 1-3 unterstützt (V.110, HDLC, T.30, ISO 7776, SDLC, LAPD, PPP, T.90NL, ISO 8208, X.25 DCE), die während einer Verbindung gewechselt werden können. So sind neben reiner Sprach- und Datenübertragung u.a. die Dienste Teletex, Fax Gruppe 2/3/4, MHS X.400, Video und X.200 nutzbar.

2.2 Struktur des CAPI

Das CAPI arbeitet als Protokoll-Demultiplexer. Es verknüpft unterschiedliche, auf der Protokollschicht 3 aufsetzende Kommunikationsanwendungen mit einem oder mehreren ISDN-Adaptern. Dabei können mehrere logische (Schicht-3-) Verbindungen auf einen physikalischen (B-) Kanal abgebildet werden (Bild 2-1).

Die Synchronisation zwischen der Anwendung und den (asynchron eintretenden) Kommunikationsereignissen erfolgt auf der Basis von Dienstprimitiven (*messages*). Diese wurden in Anlehnung an das OSI-Referenzmodell spezifiziert und ermöglichen den Austausch von Kommandos und Meldungen zwischen Anwendung und CAPI-Instanz.

Die Dienstprimitive sind betriebssystemunabhängig und haben eine sehr einfache Struktur. Sie setzen sich zusammen aus einem 8 Byte langen *header*, der die Gesamtlänge der *message* in Byte, die Applikationsnummer, das Kommando bzw. die Meldung und eine eindeutige *message*-Folgenummer enthält. Es folgt eine von dem Kommando bzw. der Meldung abhängige Anzahl von Parametern (**Bild 2-2**).

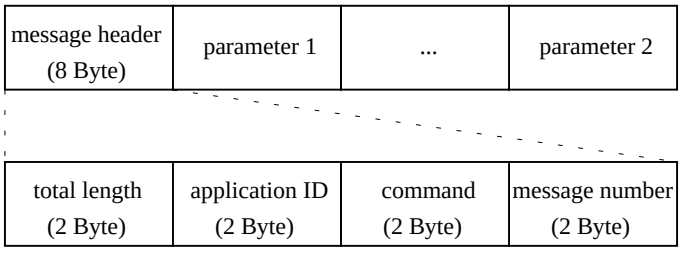


Bild 2-2: Aufbau einer CAPI-*message* [2].

Analog zum OSI-Referenzmodell werden vier Dienstprimitiv-Typen unterschieden: Das Eintreffen bestimmter Ereignisse meldet das CAPI der Anwendung mit einer *indication* (`_IND`), die von der Anwendung mit einer *response* (`_RESP`) beantwortet werden muß. Umgekehrt übergibt die Anwendung Kommandos an das CAPI mit einem *request* (`_REQ`); dieses bestätigt die Kommandoausführung mit einer *confirmation* (`_CONF`) (**Bild 2-3**).

Zusammengehörige Paare von Dienstprimitiven erkennen CAPI und Anwendung an der eindeutigen *message number*: Die zu einem *request* gehörige *confirmation* und die auf eine *indication* folgende *response* trägt jeweils dieselbe Nummer im *message header* (Bild 2-2).

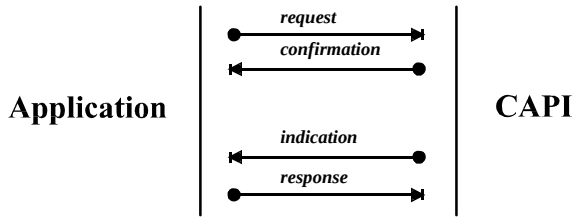


Bild 2-3: *message*-Typen zur Synchronisation von CAPI-Instanz und Anwendung

Für den Austausch der Kommandos und Meldungen werden von der CAPI-Instanz *message queues* eingerichtet. Jeder Anwendung wird bei ihrer Anmeldung (`CAPI_REGISTER`) eine eigene *queue* für *messages* von der CAPI-Instanz an die Anwendung zugewiesen (**Bild 2-4**). Dabei vergibt das CAPI der Anwendung zur eindeutigen Unterscheidung eine Applikationsnummer (Application ID); sie wird bei Abmeldung der Anwendung wieder freigegeben (`CAPI_RELEASE`).

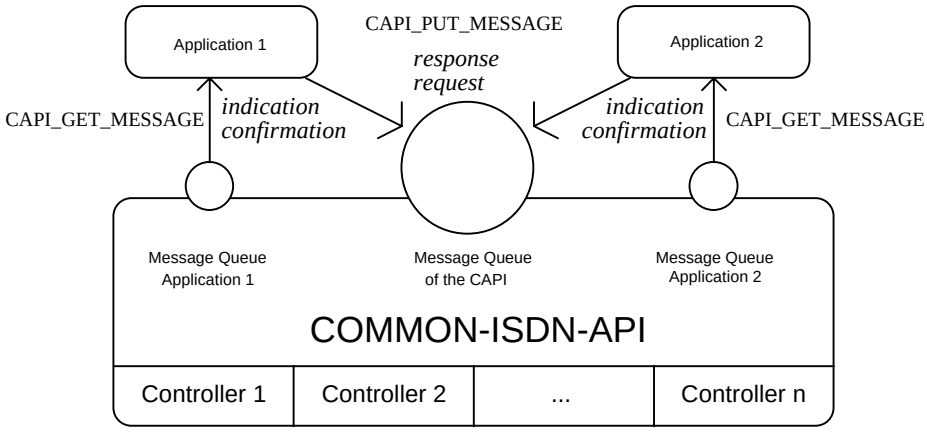


Bild 2-4: Struktur der *message queues* von CAPI und Anwendung [2].

Die Übergabe einer *message* an die CAPI-Instanz erfolgt mit der CAPI-Anweisung `CAPI_PUT_MESSAGE`. Die *message* wird dabei an eine von allen angemeldeten Anwendungen gemeinsam genutzte *message queue* der CAPI-Instanz gehängt (Bild 2-4).

Eine an eine Anwendung gerichtete *message* wird von der CAPI-Instanz in die *message queue* dieser Anwendung eingetragen. Dieser *queue* kann die *message* mit der CAPI-Anweisung `CAPI_GET_MESSAGE` entnommen werden. Das Vorhandensein neuer *messages* kann die Anwendung entweder durch aktives *polling*, d.h. ein regelmäßiges Durchsuchen der *queue*, oder durch die Anmeldung einer Signalisierungsfunktion (`CAPI_SET_SIGNAL`) feststellen, die dann nach Ablegen einer *message* in der *queue* von der CAPI-Instanz aktiviert wird.

2.3 CAPI-*messages*

Das CAPI unterscheidet drei *message*-Klassen. Klasse I umfaßt Kommandos und Meldungen zur Signalisierung. Dazu zählen der physikalische Verbindungsaufbau (`CONNECT`), eine Bereit-

schaftsmeldung an das Netz (ALERT), der Austausch von Signalisierungsinformationen (INFO) und der Abbau einer physikalischen Verbindung (DISCONNECT).

Zur Klasse II zählen alle *messages*, die sich auf die logische Verbindung beziehen: Aufbau der Schicht-3-Verbindung (CONNECT_B3), die Rücksetzung der Verbindung (RESET), der Transfer von Nutzdaten (DATA_B3) und der Abbau einer logischen Verbindung (DISCONNECT_B3). Sie setzen eine physikalische Verbindung voraus.

Die Klasse III schließlich enthält alle administrativen *messages*: das Umschalten des Adapters in den Listen-Zustand (LISTEN), die Auswahl oder Meldung spezieller Hardware-Eigenschaften (FACILITY), die Protokollwahl (SELECT_B_PROTOCOL) sowie herstellerspezifische Erweiterungen (MANUFACTURER).

Der Zusammenhang zwischen den *messages* sowie deren korrekte Abfolge z.B. beim Verbindungsaufbau sind in der CAPI-Spezifikation durch Zustandsautomaten festgelegt [2].

2.3.1 Verbindungsaufbau

Um einen ISDN-Verbindungsaufbauwunsch annehmen zu können, muß die CAPI-Instanz zunächst in den *listen*-Zustand (LISTEN_REQ) versetzt werden. Dabei kann die Annahme eintreffender Rufe auf ausgewählte Dienste beschränkt werden (z.B. Fax, Teletex, MHS).

Die Etablierung einer ISDN-Verbindung setzt sich aus zwei Schritten zusammen: dem Aufbau einer physikalischen Verbindung (B-Kanal) und dem einer logischen Verbindung.

Mit dem aktiven Aufbauwunsch für einen B-Kanal können die Kommunikationsprotokolle für die Schichten 1-3 gewählt werden (CONNECT_REQ). Als Standardeinstellung verwendet das CAPI ISO 7776 (HDLC, X.75 SLP, transparent). Die eingestellten Protokolle werden bei allen auf diesem B-Kanal aufsetzenden logischen Verbindungen verwendet. Sie können im Verlauf der Kommunikation explizit gewechselt werden (SELECT_B_PROTOCOL_REQ). Die von der jeweiligen CAPI-Instanz (bzw. der darunterliegenden Kommunikationshardware) unterstützten Protokolle können über das CAPI erfragt werden (CAPI_GET_PROFILE).

Kommt die physikalische Verbindung zustande, d.h. akzeptiert die gerufene Station den ankommenden Ruf (CONNECT_RESP), wird von den CAPI-Instanzen ein eindeutiger *physical link connection identifier* (PLCI) vergeben (CONNECT_ACTIVE_IND) (**Bild 2-5**).

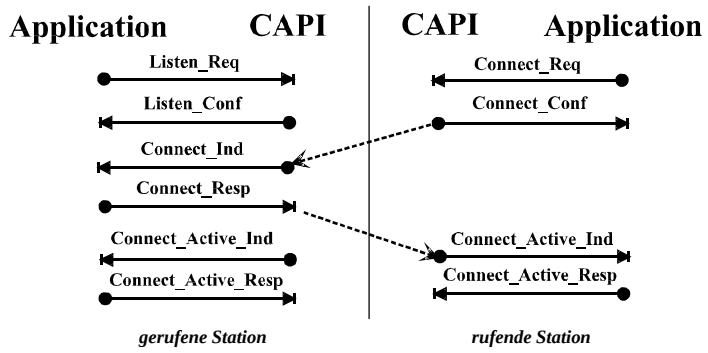


Bild 2-5: Aufbau einer physikalischen Verbindung

Nach dem erfolgreichen Aufbau einer auf einem physikalischen Kanal aufsetzenden logischen (Schicht-3-) Verbindung (CONNECT_B3_REQ) wird der Verbindung von der CAPI-Instanz eine eindeutige Kennung, der *network control connection identifier* (NCCI) zugewiesen (CONNECT_B3_ACTIVE_IND) (**Bild 2-6**).

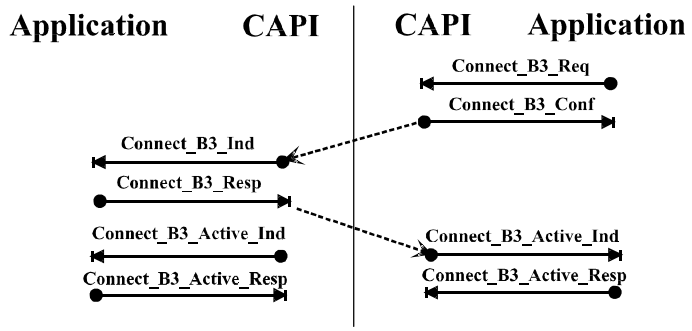


Bild 2-6: Aufbau einer logischen Verbindung

Einem physikalischen (B-) Kanal können mehrere logische Verbindungen zugeordnet werden. Sie werden durch den NCCI eindeutig identifiziert. Ihre Zahl ist nur durch den bei der Anmeldung der Anwendung (CAPI_REGISTER) reservierten Speicherplatz beschränkt.

2.3.2 Datenübertragung

Nach erfolgreichem Aufbau von physikalischer und logischer Verbindung kann die Datenübertragung beginnen. Dazu werden Datenpakete mit einem Übertragungskommando an das CAPI übergeben (DATA_B3_REQ). Sie werden von der CAPI-Instanz über den zugehörigen B-Kanal mit den beim Verbindungsaufbau gewählten Protokollen übertragen. Die empfangende CAPI-Instanz meldet eingetroffene Pakete an die Anwendung (DATA_B3_IND).

Die maximal zulässige Größe der Datenpakete hängt vom gewählten Protokoll ab. In den Protokollen der Standardeinstellung (ISO 7776) ist sie auf 128 Byte beschränkt. Es werden jedoch Protokolle unterstützt, die Paketgrößen bis 2048 Byte zulassen. Bei der Anmeldung der Anwendung gegenüber der CAPI-Instanz (CAPI_REGISTER) muß ein entsprechend großer Pufferbereich für eintreffende Pakete bereitgestellt werden [2].

2.3.3 Verbindungsabbau

Der Verbindungsabbau erfolgt entweder auf Initiative einer CAPI-Instanz (z.B. beim Auftreten von Protokollfehlern) oder wird von einem der Kommunikationspartner eingeleitet (DISCONNECT_[B3_]REQ). Auch der Abbau einer Verbindung erfolgt in zwei Teilen: Zuerst wird die logische Verbindung abgebaut (DISCONNECT_B3_IND); anschließend kann - sofern keine weitere logische Verbindung auf diesem B-Kanal besteht - die zugehörige physikalische Verbindung getrennt werden (DISCONNECT_IND).

Sowohl logische als auch physikalische Verbindungen werden erst nach Bestätigung der Verbindungsabbaumeldung durch die Anwendung (DISCONNECT_[B3_]RESP) ungültig. Erst dann können NCCI bzw. PLCI erneut von der CAPI-Instanz vergeben werden.

3 Die Sicherheitsmechanismen des S-CAPI

Um die Sicherheitsmechanismen vollständig transparent, d.h. sowohl von der eingesetzten ISDN-Hardware als auch von der darüberliegenden Kommunikationsanwendung unabhängig zu halten, wurden diese in einen Schnittstellentreiber integriert, der auf dem CAPI aufsetzt und sich gegenüber der Anwendung wiederum als CAPI darstellt (**Bild 3-1**). Dieses ***Security Common ISDN Application Programming Interface*** (**S-CAPI**) umfaßt Sicherheitsmechanismen zur Datenintegrität, Vertraulichkeit, Zugriffskontrolle und Authentikation der Kommunikationspartner (ISO 7498-2 [5]).

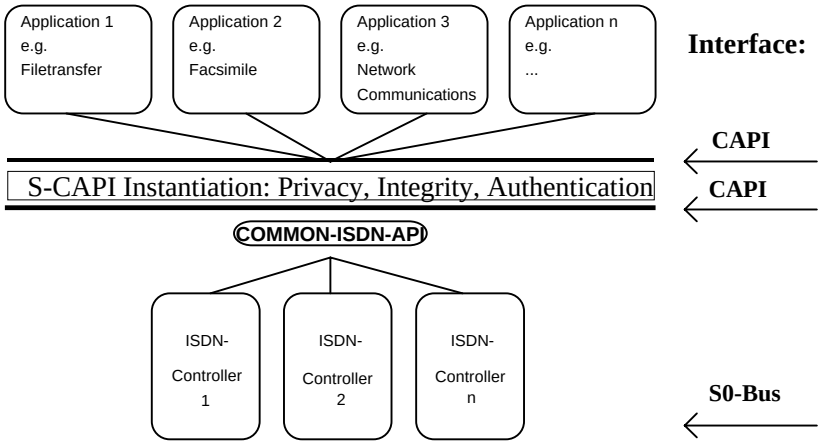


Bild 3-1: Realisierung des S-CAPI als transparenter CAPI-Schnittstellentreiber

Die Sicherheitsmechanismen des S-CAPI schützen die Nutzdaten einer ISDN-Verbindung zwischen zwei Endgeräten. Sie sind unabhängig von den durch die Applikation gewählten B-Kanal-Protokollen: Geschützt werden alle an das CAPI übergebenen Nutzdaten (`DATA_B3`). Der Schutz erfolgt dabei rufnummern- und verbindungsbezogen: Je Rufnummernpaar und Verbindung wird ein separater (einmaliger) Schlüssel verwendet.

3.1 Vertraulichkeit

Die Geheimhaltung der Nutzdaten bei der Übertragung ist ein wesentlicher Sicherheitsdienst des S-CAPI. Zu diesem Zweck werden alle auf einem B-Kanal zu übertragenen Daten mit einem symmetrischen Kryptosystem verschlüsselt. Der für die Verschlüsselung erforderliche geheime Schlüssel wird nur während einer einzigen Verbindung verwendet (*session key*). Er wird beim Verbindungsaufbau zwischen den Kommunikationspartnern, genauer: zwischen den S-CAPI-Instanzen ausgehandelt (siehe Abschnitt 3.4).

Als Verschlüsselungsverfahren kommen zwei symmetrische Blockchiffren in Frage, die der Forderung nach standardisierten und gut untersuchten Verfahren genügen: der 1977 in den USA standardisierte *Data Encryption Standard* (DES) [6] und der noch junge, von der ETH Zürich in Zusammenarbeit mit der Schweizer Firma Ascom Tech entwickelte *International Data Encryption Algorithm* (IDEA) [7, 8]. Zu Testzwecken wurde auch der von der japanischen NTT entwickelte und 1987 vorgestellte *Fast Data Enciphering Algorithm* (FEAL) ein-

gesetzt [9, 10], der eine sehr schnelle Softwareimplementierung ermöglicht, dessen Sicherheit jedoch problematisch ist.

Für Blockverschlüsselungsverfahren wurden in der internationalen Normung vier verschiedene Betriebsarten spezifiziert: Der *electronic codebook mode* (ECB), der *cipher block chaining mode* (CBC), der *output feedback mode* (OFB) und der *cipher feedback mode* (CFB) [11, 12, 13]. Diese Modi legen zentrale Eigenschaften der Verschlüsselung fest: Fehlerexpansion, Synchronisierung und die ggf. erforderliche Sonderbehandlung des letzten Datenblocks [14, 15, 16]. Im S-CAPI-Projekt wird eine 64-Bit-Blockchiffre im CBC-Modus verwendet. Dadurch ist sichergestellt, daß jedes Bit des Schlüsseltextes von allen vorausgehenden Klartextbits abhängt. Ein *replay*-Angriff wird so erheblich erschwert.

Der CBC-Modus erzeugt grundsätzlich einen Chiffretext, dessen Länge ein ganzzahliges Vielfaches der Blocklänge der Chiffre ist (hier: 64 Bit bzw. 8 Byte). Bei Nachrichten, deren Länge kein Vielfaches der Blocklänge ist, ist ein *padding* der Nachricht erforderlich. Auf diese Weise können Datenpakete aber nicht längentransparent verschlüsselt werden, wie es für die Realisierung des S-CAPI erforderlich ist. Daher wurde das Padding durch eine Sonderbehandlung des letzten Datenblocks nach einem von Davies und Price vorgeschlagenen Verfahren ersetzt [14, 16]. Dabei wird der letzte Teilblock mit den ersten Bits des zweifach verschlüsselten vorletzten Blocks XOR-verknüpft.

3.2 Integrität

Neben dem Schutz vor unbefugter Kenntnisnahme muß auch eine unbemerkte Manipulation der Nutzdaten verhindert werden. Dies kann direkt durch Anhängen einer kryptographischen Checksumme erreicht werden. Um *replay*-Angriffe, d.h. ein Wiedereinspielen von zu einem früheren Zeitpunkt abgehörter Pakete zu verhindern, sollte zusätzlich ein Zeitstempel (*time stamp*) oder eine Folgenummer eingefügt und verschlüsselt werden.

Da für die Realisierung des S-CAPI Längentransparenz gefordert wurde, wird die Integrität der übertragenen Daten indirekt sichergestellt: Durch die Verschlüsselung der Prüfsummen der verwendeten Transportprotokolle wird eine Veränderung der (verschlüsselten) Daten mit hoher Wahrscheinlichkeit vom darüberliegenden Transportprotokoll erkannt. Die Verantwortung für die Verwendung eines geeigneten Redundanz-Checkwertes und der Schutz vor *replay*-Angriffen durch die Verwendung von Folgenummern oder Zeitstempeln liegt daher bei den auf dem S-CAPI aufsetzenden, d.h. oberhalb des *network layer* liegenden Kommunikationsprotokollen oder -anwendungen.

3.3 Authentikation

Elementarer Sicherheitsmechanismus der S-CAPI ist eine gegenseitige Authentikation der Kommunikationspartner: Die ISDN-Übertragung sensibler und wichtiger Daten soll ausschließlich dann erfolgen, wenn die Identität des Kommunikationspartners zweifelsfrei festgestellt und überprüft worden ist. Nur so kann eine ISDN-Verbindung vor einem Maskerade-Angriff geschützt werden.

Die Rufnummer des Teilnehmers ist ein ungenügendes Authentikationsmerkmal, da sie prinzipiell gefälscht werden kann. Zwar ist der häufig empfohlene *call back*-Mechanismus

(Rückruf mit der Nummer des Kommunikationspartners) nur mit Aufwand umgehbar; ein Mißbrauch durch einen Benutzer, der sich unbefugt Zugang zum ISDN-Anschluß verschafft, ist damit jedoch nicht ausgeschlossen. Außerdem führt *call back* zu praktischen Schwierigkeiten, da die Gebühren vom angerufenen Kommunikationspartner übernommen werden müssen.

Die Authentikation sollte daher neben der Rufnummer an ein oder mehrere weitere eindeutige, möglichst unfälschbare Merkmale gekoppelt werden. Von der S-CAPI wird ein zusätzliches Authentikationsmerkmal verwendet, das die Kenntnis eines Paßwortes ("Wissen") und den Besitz einer Chipkarte ("Haben") erfordert.

Der Authentikationsvorgang setzt sich aus zwei Teilen zusammen: Der Authentikation des Benutzers gegenüber der S-CAPI-Instanz und der gegenseitigen Authentikation zweier S-CAPI-Instanzen.

3.3.1 Authentikation des Benutzers

Die Authentikation des Benutzers erfolgt bei der Installation der S-CAPI. Sie wird mit Hilfe eines kryptographischen *challenge response*-Protokolls unter Verwendung einer Einweg-funktion **f** durchgeführt: Nach Anforderung der Chipkarte und Eingabe der Benutzer-PIN $\mathbf{PIN_A}$ (*personal identification number*) fordert die Installationsroutine der S-CAPI eine Zufallszahl $\mathbf{r_C}$ von der Chipkarte an. Daraufhin wird ein Authentikator aus einer Zufallszahl $\mathbf{r_A}$, der Identität $\mathbf{ID_A}$, der Zufallszahl $\mathbf{r_C}$ und der $\mathbf{PIN_A}$ bestimmt und an die Chipkarte übertragen.

Die Chipkarte prüft diesen Authentikator, indem sie $\mathbf{f(r_A, ID_A, r_C, PIN'_A)}$ mit der auf der Karte geschützt abgelegten $\mathbf{PIN'_A}$ berechnet und das Ergebnis mit dem empfangenen Authentikator vergleicht [17, 15]. Ist die Authentikation erfolgreich, gibt die Chipkarte den geheimen (asymmetrischen) Schlüssel $\mathbf{SK_A}$ an die S-CAPI-Instanz zurück (**Bild 3-2**).

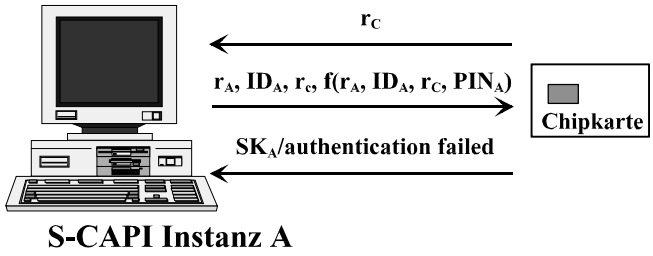


Bild 3-2: Authentikation des Benutzers gegenüber der Chipkarte

Die Aktualität dieses Authentisierungsvorgangs wird durch die Verwendung einer von der Chipkarte generierten Zufallszahl ($\mathbf{r_C}$) sichergestellt. Die Zufallszahl $\mathbf{r_A}$ verhindert, daß ein als Chipkarte maskierter Angreifer $\mathbf{r_C}$ so wählen kann, daß ein bestimmter Authentikator von dem S-CAPI berechnet wird. Der geheime Schlüssel $\mathbf{SK_A}$ wird bei der gegenseitigen Authentikation der S-CAPI-Instanzen für die Generierung eines unfälschbaren, benutzerabhängigen Authentikationsmerkmals und später für die Aushandlung eines *session keys* benötigt (Abschnitt 3.4).

Die mit der $\mathbf{PIN_A}$ des Benutzers personalisierte und geschützte Chipkarte enthält neben dem geheimen Schlüssel $\mathbf{SK_A}$ ein zugehöriges Zertifikat $\mathbf{Z_A}$ mit dem zu $\mathbf{SK_A}$ passenden öffentlichen Schlüssel $\mathbf{PK_A}$ und den öffentlichen Schlüssel $\mathbf{PK_Z}$ der Zertifizierungsinstanz, der eine Zertifikatsprüfung ermöglicht (Abschnitt 3.4).

3.3.2 Gegenseitige Authentikation der S-CAPI-Instanzen

Beim Verbindungsaufbau erfolgt der zweite Teil des Authentikationsvorgangs: eine gegenseitige Authentikation *(mutual authentication)* der S-CAPI-Instanzen, in deren Verlauf diese einander anhand unfälschbarer Authentikationsmerkmale die Identität ihres Benutzers "beweisen". Dabei kommt ein asymmetrisches *challenge response*-Protokoll nach ISO 9798-3 zur Anwendung (**Bild 3-3**), das einen parallelen Ablauf der Authentikation ermöglicht [18].

Nach erfolgreichem Verbindungsaufbau durch das CAPI wird von der aktiven, d.h. die Verbindung initiierenden Instanz (hier: **A**) das Authentikationsprotokoll durch Versendung einer Zufallszahl $\mathbf{r_A}$ und des Zertifikats $\mathbf{Z_A}$, das den öffentlichen Schlüssel $\mathbf{PK_A}$ enthält, eingeleitet.

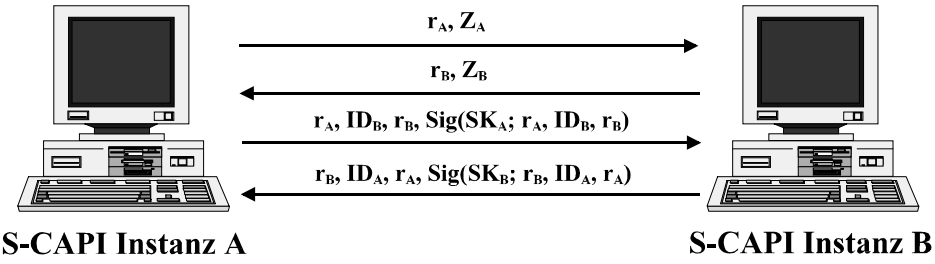


Bild 3-3: *Challenge response*-Protokoll zur gegenseitigen Authentikation der S-CAPI-Instanzen unter Verwendung eines digitalen Signatursystems

Die Instanz **B** antwortet entsprechend mit einer Zufallszahl $\mathbf{r_B}$ und dem Zertifikat $\mathbf{Z_B}$. Beide Instanzen prüfen nun mit Hilfe des ihnen bekannten öffentlichen Schlüssels $\mathbf{PK_Z}$ der Zertifizierungsinstanz, ob die Zertifikate korrekt (Signatur) und gültig sind (Zeitstempel).

Sind beide Prüfungen erfolgreich, generieren die S-CAPI-Instanzen mit ihrem geheimen Schlüssel $\mathbf{SK_A}$ ($\mathbf{SK_B}$) jeweils einen Authentikator, genauer: eine Digitale Signatur $\mathbf{Sig(SK_A; r_A, ID_B, r_B)}$ bzw. $\mathbf{Sig(SK_B; r_B, ID_A, r_A)}$ und schicken diese an die Partnerinstanz. Die kann die empfangene Signatur mit dem zuvor erhaltenen öffentlichen Schlüssel $\mathbf{PK_B}$ ($\mathbf{PK_A}$) prüfen.

3.4 Schlüsselmanagement

Für die Verschlüsselung der Nutzdaten mit einem symmetrischen Kryptosystem müssen Sender und Empfänger, hier also die S-CAPI-Instanzen der ISDN-Kommunikationspartner, über einen gemeinsamen geheimen Schlüssel verfügen, der regelmäßig gewechselt werden sollte. Die S-CAPI verwendet daher *session keys*, die nur für die Dauer einer Verbindung Gültigkeit besitzen. Sie werden beim Verbindungsaufbau zwischen den beiden S-CAPI-Instanzen vereinbart.

Für die Realisierung dieser Schlüsselvereinbarung sind klassische symmetrische Protokolle wie beispielsweise die in [19] vorgestellten ungeeignet. Sie haben den prinzipiellen Nachteil, daß sie die Vereinbarung jeweils eines gemeinsamen geheimen *master keys* zwischen jedem Paar von Kommunikationspartnern über einen sicheren (geheimen) Kanal voraussetzen. Dies würde erheblichen Speicherbedarf und Schlüsselaustauschaufwand verursachen, der zudem quadratisch mit der Zahl der Kommunikationsteilnehmer anstiege.

Aus diesem Grund wird im S-CAPI-Projekt ein asymmetrisches Schlüsselaustauschprotokoll nach CCITT X.509 bzw. ISO/IEC 9594-8 eingesetzt [20]. Dabei steigt die Anzahl der *master keys* $\mathbf{PK_i}$ nur linear mit der Zahl der Kommunikationspartner: Für jeden Benutzer genügt die Generierung eines solchen Schlüssels (mit passendem geheimen Schlüssel $\mathbf{SK_i}$), der zuvor nicht geheim ausgetauscht werden muß. Die *master keys* werden lediglich authentisch, d.h. vor Veränderung geschützt, den jeweiligen Kommunikationspartnern mitgeteilt.

Diese authentische Weitergabe erfolgt mit Hilfe von Zertifikaten $\mathbf{Z_A}$, d.h. unfälschbaren, von einer zentralen Zertifizierungsinstanz ausgestellten Gültigkeitsnachweisen. Sie enthalten den öffentlichen Schlüssel $\mathbf{PK_A}$ des Benutzers, dessen Namen $\mathbf{ID_A}$, den Namen $\mathbf{ID_Z}$ der Zertifizierungsinstanz und ein "Verfallsdatum" $\mathbf{t_A}$ und werden, wie in Abschnitt 3.3.2 beschrieben, im Rahmen des Authentikationsprotokolls ausgetauscht.

Mit einer Digitalen Signatur $\mathbf{Sig(SK_Z; PK_A, ID_A, ID_Z, t_A)}$ der Zertifizierungsinstanz wird das Zertifikat vor unbefugter Veränderung geschützt. Die Signatur (und damit die Unverfälschtheit des Zertifikats) kann mit dem zuvor authentisch veröffentlichten Prüfschlüssel $\mathbf{PK_Z}$ der Zertifizierungsinstanz verifiziert werden.

Jeder Benutzer muß nur seinen persönlichen geheimen Schlüssel $\mathbf{SK_A}$ vor unberechtigter Kenntnisnahme geschützt aufbewahren. Zu diesem Zweck wird dieser bei der Personalisierung unzugänglich auf der Chipkarte abgelegt. Mit diesem geheimen Schlüssel erfolgt die gegenseitige Authentikation zweier S-CAPI-Instanzen nach dem Verbindungsaufbau (Abschnitt 3.3.2, Bild 3-3).

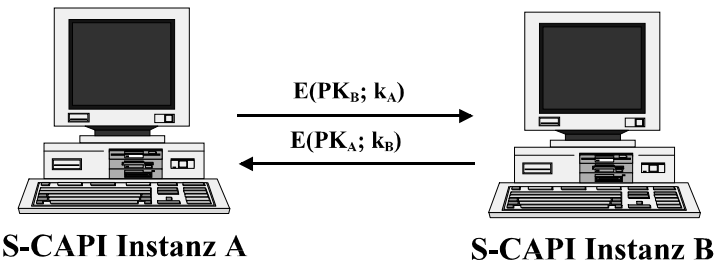


Bild 3-4: Schlüsselvereinbarung zwischen zwei S-CAPI-Instanzen

Daran schließt sich unmittelbar die Aushandlung eines *session keys* an: Ist der Kommunikationspartner erfolgreich authentisiert, generiert die aktive Station einen Zufallswert $\mathbf{k_A}$ und schickt diesen verschlüsselt an den Partner, der mit einer verschlüsselten Übermittlung einer Zufallszahl $\mathbf{k_B}$ antwortet (**Bild 3-4**). Daraus bilden beide Instanzen den *session key* $\mathbf{k_{AB}}$ durch Verknüpfung, z.B.: $\mathbf{k_{AB} = k_A \text{ XOR } k_B}$.

4 Der S-CAPI-Prototyp

Für *personal computer* wurde ein S-CAPI-Prototyp unter dem Betriebssystem DOS entwikkelt. Das CAPI für DOS, das inzwischen von jedem (deutschen) Hersteller von ISDN-Adaptern mitgeliefert wird, ist ein über einen Software-Interrupt aktivierbarer Hardwaretreiber, der die CAPI-Funktionen auf die herstellerspezifische Ansteuerung der ISDN-Hardware abbildet.

Die S-CAPI-Instanz wurde als resident installierbarer DOS-Treiber implementiert, der sich gegenüber der Kommunikationsanwendung wie ein CAPI-Treiber verhält, d.h. über denselben Software-Interrupt angesprochen wird, und seinerseits die Funktionen eines (zuvor zu installierenden) DOS-CAPI-Treibers verwendet [21].

Der S-CAPI-Treiber führt bei einem Verbindungsaufbau das Authentikationsprotokoll durch, sorgt für die Vereinbarung eines *session keys* und ver- bzw. entschlüsselt alle Datenpakete einer Verbindung. Bei der Installation fordert er zusätzlich Chipkarte und PIN zur Authentikation des Benutzers an.

Die Implementierung erfolgte auf aktiven ISDN-Karten, die über eigenes RAM verfügen. Als symmetrische Kryptosysteme kamen sehr schnelle Softwareimplementierungen von DES, IDEA und FEAL (>1 MBit/s) sowie DES-Hardware [22] zum Einsatz. Daher traten bei der Datenübertragung (64 kBit/s, keine Kanalbündelung) auch auf PCs mit 80386-Mikroprozessor keine meßbaren Geschwindigkeitseinbußen auf. Allein der Verbindungsaufbau verzögerte sich durch das zusätzliche Authentikationsprotokoll und den Signaturtest um 1-2 Sekunden. Als Digitale Signatursysteme wurden Softwareimplementierungen von RSA [23, 24], ElGamal [25] und DSS [26, 27] verwendet.

5 Fazit

Mit dem vorgestellten Konzept zur Integration von Sicherheitsmechanismen in ISDN-Kommunikationslösungen ist es durch die Verwendung des CAPI-Standards möglich, existierende Anwendungen ohne Modifikation mit den elementaren Sicherheitsdiensten Datenintegrität, Authentikation, Zugriffskontrolle und Vertraulichkeit auszustatten.

Der entwickelte S-CAPI-Prototyp belegt die Praxistauglichkeit dieses Ansatzes: Unter Verwendung bewährter, gut untersuchter kryptographischer Verfahren und standardisierter Sicherheitsprotokolle wurde eine für den Benutzer vollständig transparente und höchsten Sicherheitsanforderungen genügende Lösung realisiert.

In einer nächsten Ausbaustufe wird der S-CAPI-Prototyp um Mechanismen für ein *remote*-Sicherheitsmanagement erweitert. Ein weiterer Prototyp für UNIX-*workstations* ist geplant.

6 Dank

Detlef Dienst, Rudi Schöngarth, Dieter Schmidt und Olaf Junklewitz implementierten im Rahmen von Studien- und Diplomarbeiten wesentliche Teile des S-CAPI-Prototyps. Uwe Latsch danken wir für wertvolle Hinweise zur Gestaltung dieses Beitrags.

7 Literatur

[1] *Common ISDN Application Programming Interface*. Spezifikation des ISDN-Arbeitskreises der Deutschen Bundespost Telekom, Version 1.1, Profil A, 7.9.1990.

[2] *Common ISDN Application Programming Interface - Version 2.0*. Spezifikation des COMMON-ISDN-API-Arbeitskreises der Telekom, Projekt ROLAND, Version 2.0, Februar 1994.

[3] Heywood, Peter: *ISDN APIs Unbind Applications From Adapters*. Data Communications International, Mai 1994, S. 49-52.

[4] International Organisation for Standardization (ISO): *Open Systems Interconnection: Basic Reference Model*. International Standard ISO 7498, 1983.

[5] International Organisation for Standardization (ISO): *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture*. International Standard ISO 7498-2 (E), Genf 1989.

[6] National Bureau of Standards (NBS): *Data Encryption Standard (DES)*. Federal Information Processing Standards Publication (FIPS-PUB) 46-1, US Department of Commerce, 1/1977.

[7] Lai, Xuejia; Massey, James L.: *A Proposal for a New Block Encryption Standard*. In: Damgård, Ivan Bjerre (Hrsg.): Proceedings of Eurocrypt '90. LNCS 473, Springer, Berlin 1991, S. 389-404.

[8] Brüggemann, Theodor; Bürk, Holger: *Damit Geheimdaten vertraulich bleiben: Verschlüsselungsalgorithmus IDEA löst DES ab*. Elektronik 10/1993, S. 84-93.

[9] Shimizu, Akira; Miyaguchi, Shoij: *Fast Data Encipherment Algorithm FEAL*. In: Chaum, D.; Price, W.L. (Hrsg.): Proceedings of Eurocrypt '87, LNCS 304, Springer, Berlin 1988, S. 267-278.

[10] Miyaguchi, Shoji; Shiraishi, Akira; Shimizu, Akihiro: *Fast Data Encipherment Algorithm FEAL-8*. Review of the Electrical Communications Laboratories, Vol. 36, No. 4, 1988, S. 433-437.

[11] National Bureau of Standards (NBS): *DES modes of Operations*. Federal Information Processing Standards Publication (FIPS-PUB) 81, US Department of Commerce, 12/1980.

[12] International Organisation for Standardization (ISO): *Information processing - modes of operation for a 64-bit block cipher algorithm*. International Standard ISO 8732, Genf 1987.

[13] International Organisation for Standardization (ISO): *Modes of Operations for an N-bit Block Cipher Algorithm*. International Standard IS 10116, Genf 1991.

[14] Davies, Donald W.; Price, Wyn L.: *Security for Computer Networks*. 2. Auflage, John Wiley & Sons Ltd., Chichester 1989.

[15] Ruland, Christoph: *Informationssicherheit in Datennetzen*. DataCom-Verlag, Bergheim 1993.

[16] Fumy, Walter; Rieß, Hans Peter: *Kryptographie*. Schriftenreihe Sicherheit in der Informationstechnik, Band 6. Oldenbourg Verlag, München, 2. Auflage 1994.

[17] Fries, Otfried; Fritsch, Andreas; Kessler, Volker; Klein, Birgit: *Sicherheitsmechanismen. Bausteine zur Entwicklung sicherer Systeme*. REMO-Arbeitsberichte Bd.2, Reihe Sicherheit in der Informationstechnik, Oldenbourg, München 1993.

[18] International Organisation for Standardization (ISO): *Entity authentication mechanisms - Part 3: Entity authentication using a public-key algorithm*. Draft International Standard ISO DIS 9798-3, Genf 1992.

[19] Needham, Roger M.; Schroeder, Michael D.: *Using Encryption for Authentication in Large Networks of Computers*. Communications of the ACM, Bd. 21, Nr. 12, Dezember 1978, S. 993-999.

[20] International Organisation for Standardization (ISO): Information processing systems - *Open Systems Interconnection - The Directory - Authentication Framework.* International Standard ISO/IEC 9594-8, Genf 1989.

[21] Dienst, Detlef: *Datenverschlüsselung im ISDN unter Verwendung des Common ISDN Application Programming Interface (CAPI)*. Diplomarbeit am Institut für Nachrichtenübermittlung, Fachbereich Elektrotechnik und Informatik, Universität Siegen, 3/1994.

[22] Computer Elektronik Infosys GmbH: *SuperCrypt 99C003 Preliminary Data Sheet Vers. 1.01*. Datenblatt, 1991.

[23] Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard: *A Method for obtaining Digital Signatures and Public Key Cryptosystems*. Communications of the ACM, Bd. 21, Nr. 2, 1978, S. 120-126.

[24] Fox, Dirk: *Effiziente Softwareimplementierung asymmetrischer Kryptosysteme und der zugrundeliegenden modularen Langzahlarithmetik.* Diplomarbeit am Institut für Rechnerentwurf und Fehlertoleranz, Universität Karlsruhe, 1991.

[25] El Gamal, Taher: *A Public Key Cryptosystem an Signature Scheme Based on Discrete logarithms*. IEEE Trans. on Inform. Theory, Bd. IT-31, Nr.4, 7/1985, S. 469-472.

[26] National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication XX (FIPS-PUB), Draft, 19.8.1991.

[27] Fox, Dirk: *Der 'Digital Signature Standard': Aufwand, Implementierung und Sicherheit*. In: Weck, Gerhard; Horster, Patrick (Hrsg.): Verläßliche Informationssysteme, Proceedings der GI-Fachtagung VIS '93, Vieweg, Braunschweig 1993, S. 333-352.