In: Fumy, W.; Meister, G.; Reitenspieß, M.; Schäfer, W. (Hrsg.): "Sicherheitsschnittstellen - Konzepte, Anwendungen und Einsatzbeispiele". Proceedings des Workshops Security APIs '94, 17.-18.11.1994, München, Deutscher Universitäts-Verlag, Vieweg, Wiesbaden 1994, S. 195-210.

SECURE COMMON ISDN APPLICATION PROGRAMMING INTERFACE (S-CAPI)

Dirk Fox, Christoph Ruland

ZUSAMMENFASSUNG

Mit wachsender Bedeutung der Datenübertragung in privaten und öffentlichen Netzen spielen Sicherheitsüberlegungen eine immer wichtigere Rolle. Das Projekt S-CAPI verfolgt das Ziel, rechnerbasierte ISDN-Kommunikationslösungen transparent mit kryptographisch starken Schutzmechanismen auszustatten. Zu diesem Zweck wurden Sicherheitsmechanismen zur Modifikationserkennung, Authentikation und Vertraulichkeit für eine gesicherte ISDN-Kommunikation spezifiziert, die auf dem Herstellerstandard *Common ISDN API* (CAPI) aufsetzen.

Die Sicherheitsdienste des S-CAPI sorgen für eine gegenseitige Authentikation der Kommunikationspartner, ermöglichen eine symmetrische *online*-Verschlüsselung mit wechselnden *session keys* und umfassen eine tokenbasierte Benutzerauthentikation. Die Integration erfolgt vollständig transparent, d.h. erfordert weder eine Modifikation der ISDN-Anwendung noch des Hardware-Treibers. Die Modul- und Schnittstellenspezifikation des S-CAPI wird vorgestellt.

1. EINLEITUNG

Die Datenübertragung in öffentlichen Netzen spielt vor allem beim Einsatz moderner Kommunikationstechnik in größeren und mittleren Unternehmen eine immer wichtigere Rolle. In einer solchen kommunikationstechnischen Umgebung bekommen Fragen nach dem Schutz der übertragenen Daten große Bedeutung: So werden immer häufiger sensible Daten elektronisch übertragen, während zugleich die Kontrolle über die verwendeten Übertragungsmedien abnimmt (Vermittlungsrechner, Richtfunk-, Mobilfunk- und Satellitenkanäle). Das diensteintegrierende digitale öffentliche Netz ISDN nimmt bei dieser Entwicklung eine Schlüsselrolle ein, da es eine sehr schnelle und kostengünstige Übertragung beliebiger digitaler Daten (64 kBit/s pro Kanal) über bestehende Teilnehmeranschlußleitungen ermöglicht. 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 Sicherheitsdienste Modifikationserkennung, Authentikation und Vertraulichkeit 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 Sicherheitsmechanismen wurden die folgenden Anforderungen gestellt:

- Die Integration sollte transparent erfolgen, d.h. unabhängig sowohl von der auf der CAPI aufsetzenden Kommunikationsanwendung als auch von der eingesetzten Hardware.
- Die Spezifikation der Sicherheitsmechanismen sollte weitgehend unabhängig von speziellen kryptographischen Algorithmen und Protokollen gehalten werden.
- Die Modulstruktur und internen Schnittstellen sollten möglichst allgemein spezifizert werden, um eine Übertragung auf andere Kommunikationsschnittstellen zu ermöglichen.

2. DAS *COMMON ISDN API* (CAPI)

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 [1]. 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.

Der CAPI-Standard besteht 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 Telecommunication Union* (ITU) als Standardisierungsvorschlag eingereicht [6].

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. So ist die Schnittstelle beispielsweise unabhängig davon, ob von der Hardware ein Primär-Multiplex- (30 B-Kanäle) oder ein Basisanschluß (2 B-Kanäle) bedient wird.

Aus der Sicht des OSI-Referenzmodells [8] liegt das CAPI oberhalb des *network layers*, d.h. es bietet darüberliegenden ISDN-Anwendungen transparente Ende-zu-Ende-Verbindungen auf einem B-Kanal (Bild 1).

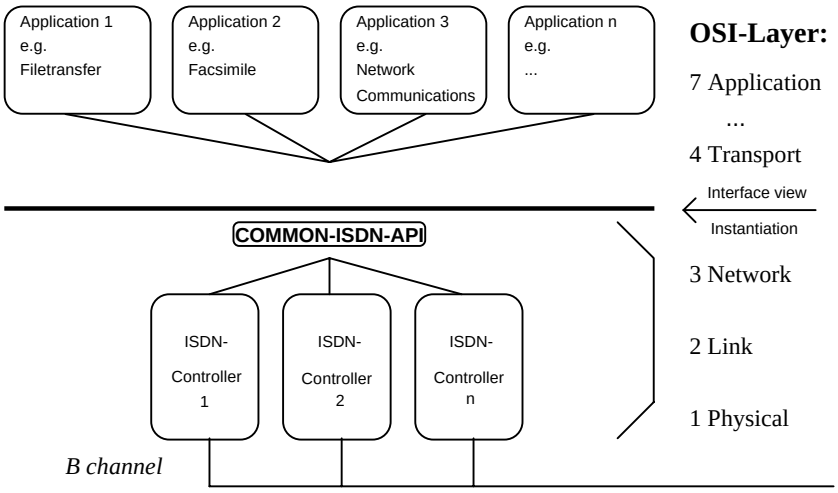

Bild 1: Einordnung des CAPI im OSI-Referenzmodell (nach [2]).

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). Sie werden beim Verbindungsaufbau gewählt und können auch während einer Verbindung gewechselt werden. Neben reiner Sprach- und Datenübertragung sind so u.a. die Dienste Teletex, Fax Gruppen 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 bei Verwendung geeigneter Protokolle auch mehrere logische (Schicht-3-) Verbindungen auf einen physikalischen (B-) Kanal abgebildet werden (Bild 1).

Die CAPI-Instanz wird von der Anwendung mit CAPI-Anweisungen (Präfix `CAPI_`) angesprochen. Die Synchronisation zwischen der Anwendung und den (asynchron eintretenden) Kommunikationsereignissen erfolgt mit Hilfe 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 *messages* sind betriebssystemunabhängig und haben eine sehr einfache Struktur. Sie setzen sich zusammen aus einem 8 Byte langen *header* und einer von dem jeweiligen Kommando bzw. der Meldung abhängigen Anzahl von Parametern.

Analog zum OSI-Referenzmodell werden vier *message*-Typen unterschieden: Das Eintreffen von Kommunikationsereignissen meldet die CAPI-Instanz der Anwendung mit einer *indication* (Postfix `_IND`, Meldung), die von der Anwendung mit einer *response* (`_RESP`) beantwortet wird. Umgekehrt übergibt die Anwendung Befehle an den CAPI-Treiber mit einem *request* (`_REQ`, Kommando); dieser bestätigt die Ausführung mit einer *confirmation* (`_CONF`) (Bild 2). Zusammengehörige *message*-Paare erkennen CAPI und Anwendung an einer eindeutigen *message number*.

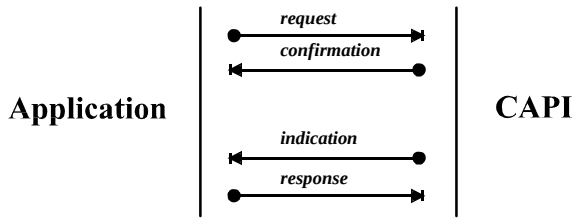

Bild 2: *message*-Typen zur Synchronisation

Für den Austausch der Kommandos und Meldungen werden von der CAPI-Instanz *message queues* eingerichtet. Jeder Anwendung weist die CAPI-Instanz bei ihrer Anmeldung (`CAPI_REGISTER`) eine eigene *message queue* zu. Alle an diese Anwendung gerichteten *messages* (`_CONF`, `_IND`) werden von der CAPI-Instanz in diese *message queue* eingetragen. Sie können mit der CAPI-Anweisung `CAPI_GET_MESSAGE` der *queue* entnommen werden (Bild 3).

Die Auslieferung einer *message* an die CAPI-Instanz (`_REQ`, `_RESP`) erfolgt mit der CAPI-Anweisung `CAPI_PUT_MESSAGE`. Die *message* wird dabei über eine von allen angemeldeten Anwendungen gemeinsam genutzte *message queue* der CAPI-Instanz übergeben.

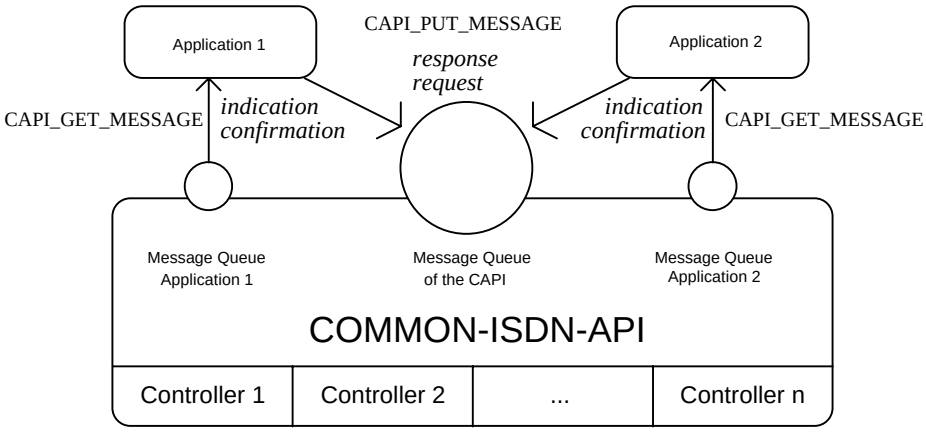

Bild 3: Struktur der *message queues* (nach [2]).

Das Vorliegen neuer *messages* in ihrer *queue* kann die Anwendung entweder durch aktives *polling*, d.h. ein regelmäßiges Durchsuchen der *queue* feststellen, oder durch Anmeldung einer Signalisierungsfunktion (`CAPI_SET_SIGNAL`) von der CAPI-Instanz anzeigen lassen.

2.3. CAPI-*messages*

Die CAPI-Spezifikation unterscheidet drei *message*-Klassen. Klasse I umfaßt Kommandos und Meldungen zur Signalisierung. Dazu zählen der physikalische Verbindungsaufbau (`CONNECT`), eine Bereitschaftsmeldung 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 etablierte 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 sind in der CAPI-Spezifikation durch Zustandsautomaten festgelegt [2].

3. DIE SICHERHEITSMECHANISMEN DES S-CAPI

Das *Security Common ISDN Application Programming Interface* (S-CAPI) umfaßt Sicherheitsmechanismen zur Datenintegrität, Vertraulichkeit, Zugriffskontrolle und Authentikation der Kommunikationspartner (ISO 7498-2 [9]). Um diese Sicherheitsmechanismen vollständig transparent, d.h. sowohl von der eingesetzten ISDN-Hardware als auch von der

darüberliegenden Kommunikationsanwendung unabhängig zu halten, wurden sie als ein auf dem CAPI aufsetzender Schnittstellentreiber realisiert, der sich gegenüber der Anwendung wiederum als CAPI darstellt (Bild 4).

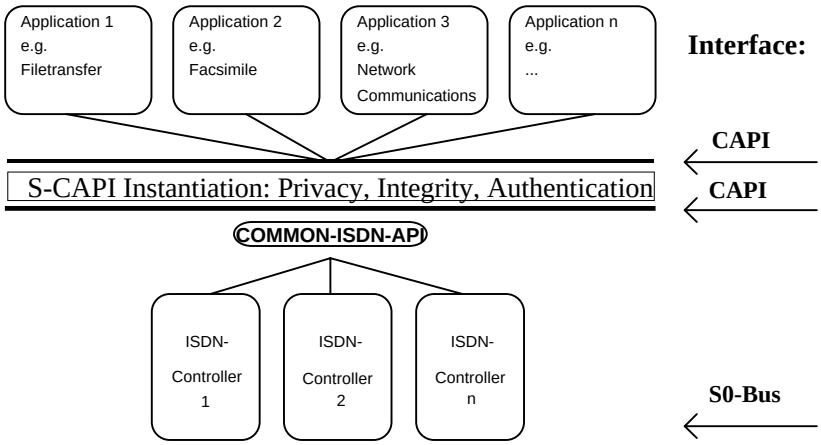

Bild 4: Transparente Realisierung des S-CAPI

Die Sicherheitsmechanismen des S-CAPI schützen die Nutzdaten einer ISDN-Verbindung zwischen zwei Endgeräten. Sie sind unabhängig von den verwendeten B-Kanal-Protokollen und Diensten: Geschützt werden alle an das CAPI übergebenen Nutzdaten (`DATA_B3`).

3.1. Vertraulichkeit

Die Geheimhaltung der Nutzdaten bei der Übertragung ist ein wesentlicher Sicherheitsdienst des S-CAPI. Zu diesem Zweck werden alle 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 (Abschnitt 3.4).

3.2. Integrität

Neben dem Schutz vor unbefugter Kenntnisnahme soll auch eine unbemerkte Manipulation der Nutzdaten verhindert werden. Dies wird üblicherweise durch Anhängen einer kryptographischen Checksumme erreicht. 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 [3].

Da für die Realisierung des S-CAPI Längentreue erforderlich ist (Abschnitt 4.2.3), kann die Integrität der übertragenen Daten nur indirekt sichergestellt werden: durch die Verschlüsselung von Zeitstempeln oder Prüfsummen der auf dem S-CAPI aufsetzenden, d.h. oberhalb des *network layer* liegenden Kommunikationsprotokolle oder -anwendungen.

3.3. Authentikation

Elementarer Sicherheitsmechanismus des S-CAPI ist eine gegenseitige Authentikation der Kommunikationspartner (*peer authentication*): 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.

Aus diesem Grund sollte die Authentikation neben der Rufnummer an andere eindeutige, möglichst unfälschbare Merkmale geknüpft werden, die im Rahmen eines kryptographischen Authentikationsprotokolls zwischen den S-CAPI-Instanzen ausgetauscht werden. Die Merkmale sollten geschützt auf einem Sicherheitstoken (z.B. einer Chipkarte) aufbewahrt und nur einem authentisierten Benutzer zugänglich sein.

3.3.1. Authentikation des Benutzers

Die Authentikation des Benutzers erfolgt bei der Installation des S-CAPI. Sie wird mit Hilfe eines kryptographischen *challenge response*-Protokolls durchgeführt. Nach Anforderung des Sicherheitstokens und Eingabe einer Benutzer-PIN PIN_A (*personal identification number*) wird ein einseitiges Authentikationsprotokoll zwischen S-CAPI-Instanz und Sicherheitstoken durchgeführt [5, 13]. Der Sicherheitstoken enthält die für das asymmetrische, zertifikatsbasierte gegenseitige Authentikationsprotokoll zwischen S-CAPI-Instanzen erforderlichen Schlüssel und Zertifikate.

3.3.2. 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 die Instanzen einander anhand unfälschbarer Authentikationsmerkmale ihre Identität "beweisen". Dabei kommt ein asymmetrisches *challenge response*-Protokoll zur Anwendung (z.B. nach ISO 9798-3 [11]; Bild 5).

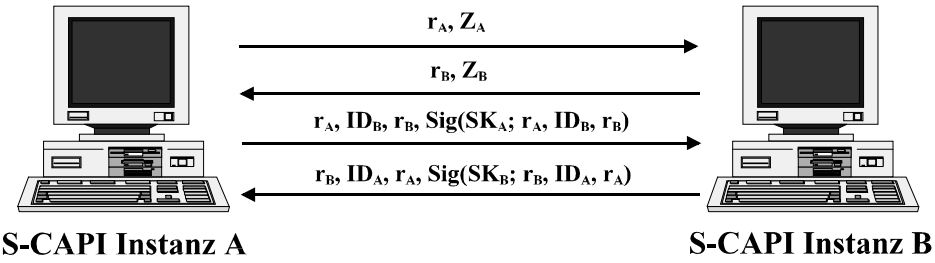

Bild 5: Beispiel eines asymmetrischen c*hallenge response*-Protokolls [13]

3.4. Schlüsselvereinbarung

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. Das S-CAPI verwendet daher *session keys*, die nur für die Dauer einer Verbindung Gültigkeit besitzen. Sie werden nach der erfolgreichen Authentikation zwischen den beiden S-CAPI-Instanzen vereinbart.

Für die Realisierung dieser Schlüsselvereinbarung sind klassische symmetrische Protokolle wie beispielsweise die in [12] vorgestellten ungeeignet. Sie haben den prinzipiellen Nachteil, daß sie je Paar von Kommunikationspartnern die Kenntnis eines gemeinsamen geheimen, über einen sicheren (geheimen) Kanal ausgetauschten *master keys* voraussetzen. Dies würde erheblichen Speicherbedarf und Schlüsselaustauschaufwand verursachen, der zudem quadratisch mit der Zahl der Kommunikationsteilnehmer anstiege.

Aus diesem Grund führt die S-CAPI-Instanz ein asymmetrisches Schlüsselaustauschprotokoll durch (z.B. nach ISO/IEC 9594-8 [10]oder [4, 7]). Dabei steigt die Anzahl der (öffentlichen) *master keys* PK_i nur linear mit der Zahl der Kommunikationspartner: Für jeden Benutzer genügt die Generierung eines solchen Schlüssels (mit passendem geheimen Schlüssel SK_i), der zuvor lediglich authentisch, d.h. vor Veränderung geschützt, den jeweiligen Benutzern mitgeteilt werden muß. Dies geschieht mit Hilfe des Sicherheitstokens und eines darin abgelegten Schlüsselzertifikats, d.h. eines von einer Authentikationszentrale ausgestellten Gültigkeitsnachweises.

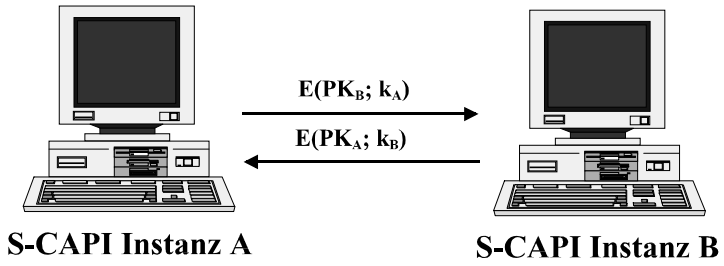

Bild 6: Schlüsselvereinbarung zwischen zwei S-CAPI-Instanzen

Bei der Schlüsselaushandlung tauschen die S-CAPI-Instanzen asymmetrisch verschlüsselt zufällig generierte Schlüsselteile k_A und k_B aus (Bild 6). Daraus bilden beide Instanzen durch eine geeignete Verknüpfung den (symmetrischen) *session key* k_{AB}.

4. INTEGRATION DER SICHERHEITSMECHANISMEN

Von einer auf dem CAPI aufsetzenden, transparenten Integration der in Kapitel 3 vorgestellten Sicherheitsmechanismen sind neben der Installation der S-CAPI-Instanz der Auf- und Abbau physikalischer und logischer Verbindungen und die Datenübertragung betroffen.

In den folgenden Abschnitten werden Struktur und Arbeitsweise des in mehrere Module aufgeteilten S-CAPI-Treibers vorgestellt. Dabei werden zwei Phasen unterschieden: die Installation (*installation phase*), die eine Benutzerauthentikation und die Konfiguration der S-CAPI-Instanz umfaßt, und die Arbeitsphase (*working phase*), in der rufnummernabhängig für Kom-

munikationskontrolle, Authentikation der Kommunikationspartner und Schutz der übertragenen Daten gesorgt wird. Besonderes Gewicht liegt auf der Beschreibung der Modulschnittstellen.[1] Es wird skizziert, wie die Sicherheitsmechanismen in der *working phase* in den Ablauf der Synchronisation von ISDN-Anwendung und CAPI eingepaßt werden.

4.1. Die Installationsphase

Die Installation einer S-CAPI-Instanz umfaßt neben dem Laden des S-CAPI-Schnittstellentreibers zwei weitere Vorgänge: die Authentikation des Benutzers und die Konfiguration mit Angaben, die eine rufnummernabhängige Unterscheidung von geschützten, ungeschützten und nicht zulässigen ISDN-Verbindungen ermöglichen (Kommunikationskontrolle). Alle Konfigurationsdaten und die nach erfolgreicher Benutzerauthentikation erhaltenen Schlüssel werden in eine zentrale *security information base* (SIB) eingetragen (Bild 7).

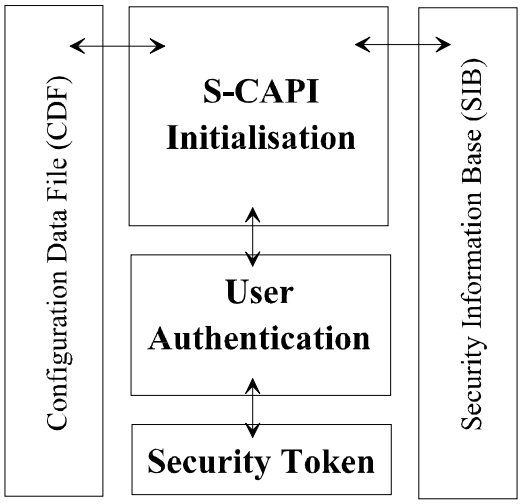

Bild 7: Installationsphase der S-CAPI-Instanz

Analog werden bei der Deinstallation der S-CAPI alle benutzerbezogenen Schlüssel und die Konfiguration durch mehrfaches Überschreiben aus der *security information base* (SIB) gelöscht, bevor die S-CAPI-Instanz entladen wird.

4.1.1. Benutzerauthentikation

Zur Authentikation des Benutzers wird ein Sicherheitstoken (*security token*), z.B. eine Chipkarte, als Berechtigungsnachweis gefordert. Gegenüber dem Sicherheitstoken muß sich der Benutzer in einem einseitigen (symmetrischen oder asymmetrischen) Authentikationsprotokoll durch die Eingabe seiner *personal identification number* (PIN) ausweisen.[2]

Nach erfolgreicher Benutzerauthentikation gibt der Sicherheitstoken den Zugriff auf die für das Authentikationsprotokoll und die Schlüsselvereinbarung erforderlichen asymmetrischen

---

1 Die Darstellung erfolgt in C-Notation. Dabei werden zur Vereinfachung weder die interne Fehlerbehandlung noch die verwendeten Datenstrukturen betrachtet.

2 Hier können unterschiedliche Sicherheitstoken und Authentikationsprotokolle Verwendung finden.

Schlüssel und Zertifikate frei. Sie werden in die globale *security information base* (SIB) der Arbeitsstation eingetragen. Schlägt die Authentikation des Benutzers fehl, wird der Installationsvorgang abgebrochen.

4.1.2. Konfiguration

Jede S-CAPI-Instanz benötigt als Konfigurationsdaten eine Liste der Rufnummern aller ISDN-Kommunikationsteilnehmer, mit denen eine geschützte Kommunikation (`USERTYPE = Secure`) erfolgen soll. Zu diesem Zweck werden diese Teilnehmer mit ihren Rufnummern in eine Konfigurationsdatei (*configuration data file*, CDF) eingetragen (Bild 8).[3]

Soll die Kommunikation mit bestimmten Teilnehmern ausgeschlossen werden, können diese explizit eingetragen werden (`USERTYPE = Illegal`). Auch Klartextverbindungen können zugelassen werden (`USERTYPE = Clear`).

```
[user]
          ISDNNO = 0049261123456
          USERTYPE = Illegal|Clear|Secure
```

Bild 8: Beispieleintrag in der S-CAPI-Konfigurationsdatei

Die Behandlung von unbekannten, d.h. nicht einzeln konfigurierten ISDN-Verbindungen während der *working phase* wird durch den "Schalter" `ALLOW_UNKNOWN_CONNECTIONS` festgelegt (Bild 9): Ist `ON` gewählt, erfolgt die Kommunikation mit unbekannten Teilnehmern im Klartext; anderenfalls wird der Aufbau einer solchen Verbindung durch die Kommunikationskontrolle verhindert (Voreinstellung: `OFF`, Abschnitt 4.2.2).

```
[configuration]
          ALLOW_UNKNOWN_CONNECTIONS = ON|OFF
```

Bild 9: Konfiguration der Behandlung unbekannter Verbindungen

Die Konfigurationsdatei (CDF) wird bei der Installation der S-CAPI-Instanz ausgelesen. Um die Konfigurationsdatei vor Manipulationen zu schützen, wird sie von der Zertifizierungsinstanz mit einer Digitalen Signatur versehen. Diese wird mit dem dem Sicherheitstoken entnommenen öffentlichen Schlüssel der Zertifizierungsinstanz geprüft. Scheitert die Signaturprüfung, wird der Installtionsvorgang abgebrochen. Anderenfalls werden die Rufnummern und zugehörigen Typ-Einstellungen in der SIB abgelegt:

```
void set_peer_type (char *isdnno, int type);
```

4.2. Die Arbeitsphase

Die Arbeitsphase des S-CAPI-Treibers umfaßt drei Einheiten: den Aufbau einer Verbindung, in dessen Rahmen die Authentikation der S-CAPI-Instanzen und die Aushandlung eines *session keys* erfolgt, die (geschützte) Kommunikation und den Verbindungsabbau, bei dem für eine Austragung der *session*-bezogenen Parameter aus der SIB gesorgt werden muß.

---

3 Die Codierung der Rufnummern erfolgt dabei nach ETS 300 102-1 / Q.931.

4.2.1. Das Modulkonzept des S-CAPI

Die S-CAPI-Instanz zerfällt in vier Module, die unterschiedliche Sicherheitsmechanismen realisieren. Diese Module wurden sowohl voneinander als auch von den verwendeten kryptographischen Algorithmen und Protokollen weitgehend unabhängig gehalten (Bild 10).

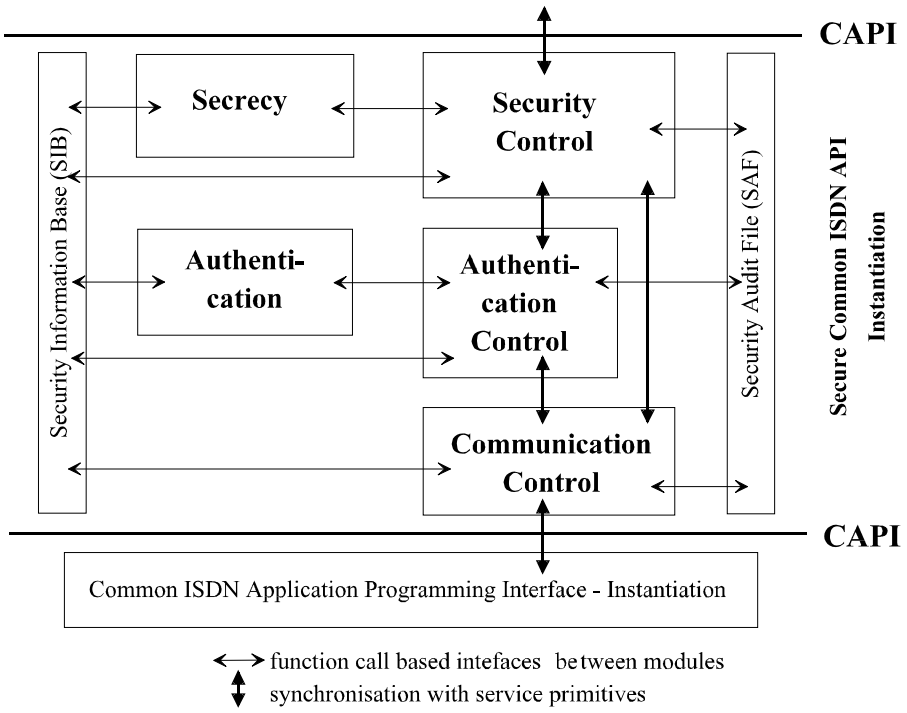

Bild 10: Modulkonzept des S-CAPI

Alle sensiblen Daten und Parameter (*session keys*, Schlüssel, Zertifikate etc.) werden in der bei der Installation der S-CAPI-Instanz angelegten *security information base* (SIB) verwaltet. Sicherheitsrelevante Vorgänge werden in einem *security audit file* (SAF) protokolliert.

4.2.2. Verbindungsaufbau

Die Etablierung einer ISDN-Verbindung erfolgt in zwei Schritten: Zunächst wird eine physikalische Verbindung (B-Kanal) aufgebaut, bei der die Kommunikationsprotokolle für die Schichten 1-3 gewählt werden.[4] Anschließend können auf diesem B-Kanal logische Verbindungen aufgesetzt werden.

Physikalischer Verbindungsaufbau

Kommt eine physikalische Verbindung zustande, d.h. akzeptiert die gerufene Anwendung einen ankommenden Ruf (`CONNECT_RESP`), wird von den CAPI-Instanzen ein eindeutiger *physical link connection identifier* (PLCI) vergeben. Dieser wird an die darüberligende S-CAPI-Instanz gemeldet (`CONNECT_ACTIVE_IND`). Erhält das S-CAPI-Modul *communication control* diese Meldung, muß es sicherstellen, daß die Verbindung nur in zulässiger Weise zustande kommt

---

4 Als Standardeinstellung verwendet das CAPI ISO 7776 (HDLC, X.75 SLP, transparent). Die eingestellten Protokolle können im Verlauf der Kommunikation gewechselt werden (`SELECT_B_PROTOCOL_REQ`).

(Kommunikationskontrolle). Dazu erfragt es den zu der Rufnummer des Anrufers gehörigen Typ-Eintrag in der SIB:

```
int get_peer_type(char *isdnno);
```

Die Funktion liefert einen von vier möglichen Werten zurück: `Clear` (Klartextkommunikation zugelassen), `Illegal` (keine Kommunikation erlaubt), `Secure` (nur geschützte Kommunikation) oder `Unknown` (d.h. kein Eintrag mit der übergebenen Rufnummer vorhanden).

Auf ein `Clear` wird die Meldung `CONNECT_ACTIVE_IND` über die Module *authentication control* und *security control* an die Anwendung weitergereicht. Die Rückgabe `Illegal` zeigt an, daß ein Verbindungsaufbau unzulässig ist: Das Modul sorgt mit dem Kommando `DISCONNECT_REQ` für einen Verbindungsabbau. Der Vorfall wird in das *security audit file* (SAF) eingetragen:

```
void report_illegal_connect (char *isdnno);
```

Wird der Typ `Secure` zurückgegeben, dann sorgt das *communication control* Modul für den Eintrag des vergebenen PLCI in der SIB, bevor die Meldung `CONNECT_ACTIVE_IND` an die Anwendung weitergereicht wird (Bild 11):

```
void put_plci(char *isdnno, unsigned long plci);
```

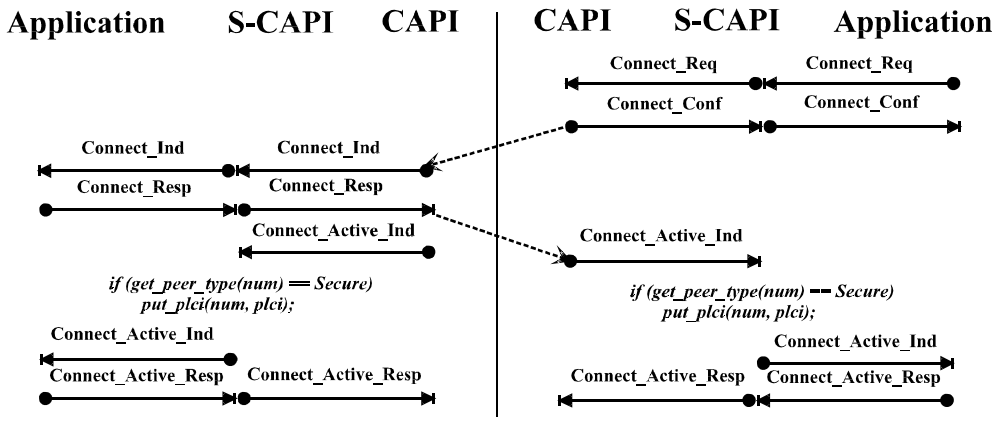

Bild 11: Aufbau einer physikalischen Verbindung (Typ: `Secure`)

Die Reaktion auf den Rückgabewert `Unknown` hängt von dem Konfigurations-"Schalter" `ALLOW_UNKNOWN_CONNECTIONS` ab: Ist dieser auf `ON` gesetzt, wird die Verbindung für Klartextkommunikation (`Clear`) zugelassen, anderenfalls wird sie wie eine unzulässige (`Illegal`) behandelt.

Logischer Verbindungsaufbau

Ist eine physikalische Verbindung etabliert, kann die Anwendung darauf eine logische Verbindung aufsetzen (`CONNECT_B3_REQ`). Kommt sie zustande, ordnen die CAPI-Instanzen dieser eine eindeutige Kennung zu, den *network control connection identifier* (NCCI), und melden diesen der darüberliegenden S-CAPI-Instanz (`CONNECT_B3_ACTIVE_IND`).

Die Meldung wird vom S-CAPI-Modul *communication control* an das Modul *authentication control* weitergereicht (Bild 10). Dort wird zunächst der Typ der angegebenen Verbindung geprüft:

```
int get_connection_type (unsigned long plci);
```

Die Funktion liefert einen von zwei möglichen Werten zurück: `Secure` (nur geschützte Kommunikation zugelassen) oder `Clear` (Klartextkommunikation erlaubt). Liegt eine Klartextverbindung vor, wird die Meldung `CONNECT_B3_ACTIVE_IND` direkt an die ISDN-Anwendung weitergegeben. Im Falle einer geschützten Verbindung (Typ `Secure`) wird der NCCI zunächst in der SIB vermerkt:

```
void put_ncci (unsigned long plci, unsigned long ncci);
```

Bevor die ISDN-Anwendung über der Abschluß des Verbindungsaufbaus informiert werden darf, ist zunächst noch eine Authentikation der Gegenstelle erforderlich. Daher wird der Status der Verbindung auf `Authenticate` gesetzt.

```
void set_session_type (unsigned long ncci, int type);
```

Authentikation des Kommunikationspartners

Vom Modul *authentication control* wird nun, wie in Abschnitt 3.3.2 vorgestellt, ein asymmetrisches *challenge-response*-Protokoll zur gegenseitigen Authentikation der S-CAPI-Instanzen eingeleitet. Dazu generiert das Modul *authentication* die erforderlichen Protokolldateneinheiten und sendet sie an die Gegenstelle; eintreffende Pakete (`DATA_B3_IND`) werden, solange die Funktion:

```
int get_session_type (unsigned long ncci);
```

den Wert `Authenticate` liefert, vom Modul *authentication control* abgefangen und zur Auswertung an das Modul *authentication* übergeben.

Schlägt die Authentikation fehl, initiiert das Modul *authentication control* einen Verbindungsabbau (`DISCONNECT_B3_REQ`, `DISCONNECT_REQ`), löscht den NCCI-Eintrag in der SIB und läßt den Vorgang im *security audit file* (SAF) dokumentieren:

```
void remove_ncci (unsigned long ncci);
void report_authentication_failed (char *isdnno);
```

Im Anschluß an eine erfolgreiche Authentikation ist, wie in Abschnitt 3.4 beschrieben, noch ein symmetrischer *session key* auszuhandeln. Das Modul *authentication control* setzt den Status der Verbindung mit `set_session_type(ncci, type)` auf `Keyexchange`.

Schlüsselvereinbarung

Die Schlüsselvereinbarung übernimmt das Modul *security control*. Dazu sind zwei verschlüsselte Zufallszahlen zwischen den beiden S-CAPI-Instanzen auszutauschen. Der *session key* wird abschließend in die SIB eingetragen:

```
void put_session_key (unsigned long ncci, void *key);
```

Schließlich wird der *session*-Typ auf `Secure` gesetzt.

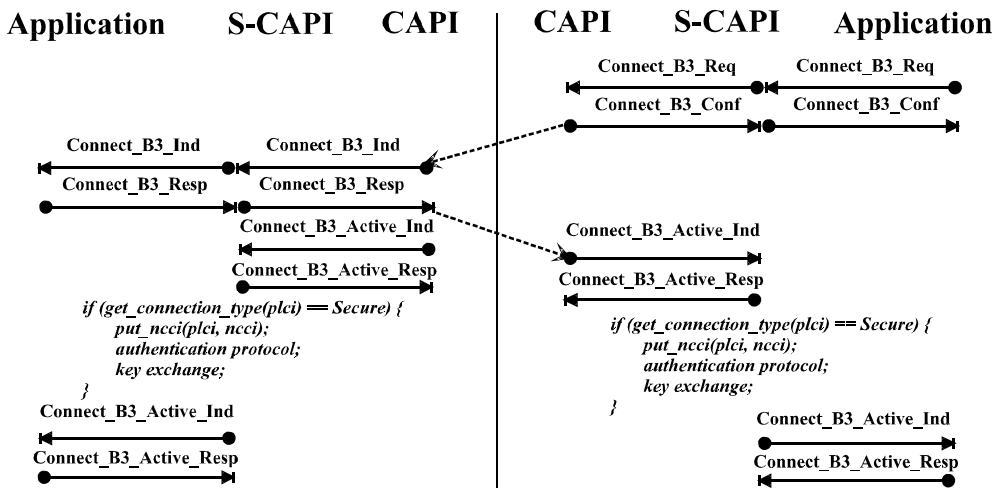

Bild 12: Aufbau einer logischen Verbindung (Typ: `Secure`)

4.2.3. Datenübertragung

Die Datenübertragung erfolgt über den zu einer logischen Verbindung (*session*) gehörigen B-Kanal mit den beim Verbindungsaufbau gewählten Protokollen. Sie wird von dem S-CAPI-Modul *security control* überwacht. Erhält es von der Anwendung einen Sendewunsch (`DATA_B3_REQ`), der einen Zeiger auf die zu übertragenen Daten und den NCCI der *session* umfaßt, wird zunächst mit `get_session_type(ncci)` der Typ der Verbindung geprüft. Liegt eine Klartextverbindung (`Clear`) vor, wird der Sendewunsch unverändert über das *communication control*-Modul an die CAPI-Instanz weitergereicht.

Handelt es sich um eine geschützte Verbindung (`Secure`), dann ist das Datenpaket vor Versendung zu verschlüsseln (Bild 13). Der dazu erforderliche *session key* wird der SIB entnommen. Damit erfolgt eine längentreue Verschlüsselung des Datenpakets.

```
void *get_session_key (unsigned long ncci);
void encrypt (void *key, char *data, int len);
```

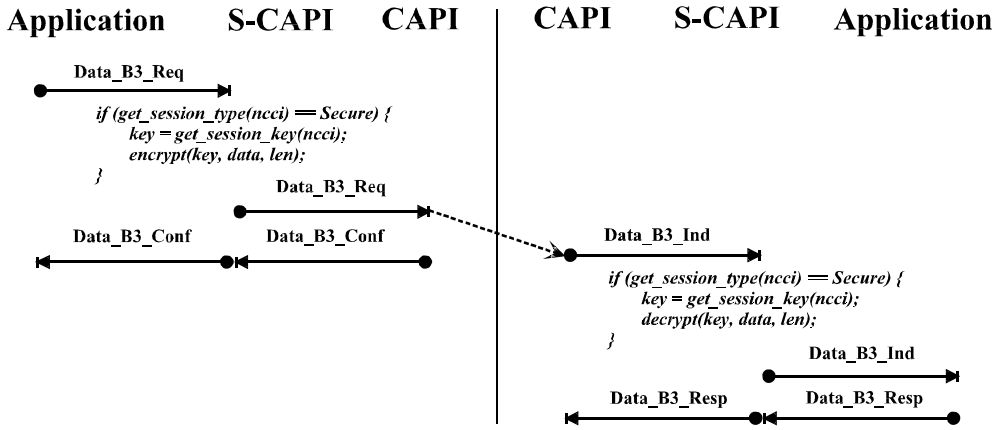

Bild 13: Geschützte Übermittlung eines Datenpaketes

Sende- und der Empfangspuffer werden von der Anwendung bzw. der CAPI-Instanz verwaltet. Daher müssen die verschlüsselten Daten wieder in demselben Puffer abgelegt werden. Die

Verschlüsselung muß zudem längentreu erfolgen, um sicherzustellen, daß Puffergrenzen und maximale Paketlängen nicht überschritten werden.

Erhält die S-CAPI-Instanz umgekehrt über das CAPI eine Empfangsmeldung (`DATA_B3_IND`), sorgt das *security control*-Modul analog für eine Entschlüsselung des empfangenen Datenpakets, wenn die Prüfung des *session*-Typs `Secure` liefert:

```
void decrypt (void *key, char *data, int len);
```

Bei Klartextverbindungen (*session*-Typ `Clear`) wird die Empfangsmeldung direkt an die ISDN-Anwendung weitergereicht.

4.2.4. Verbindungsabbau

Der Verbindungsabbau erfolgt entweder auf Initiative einer CAPI-Instanz (z.B. beim Auftreten von Protokollfehlern) oder wird von einem der Kommunikationspartner eingeleitet. Auch der Abbau einer Verbindung erfolgt in zwei Teilen: Zuerst wird die logische Verbindung abgebaut (`DISCONNECT_B3_REQ`); anschließend kann - sofern keine weitere logische Verbindung auf diesem B-Kanal besteht - die zugehörige physikalische Verbindung getrennt werden (`DISCONNECT_REQ`).

Sowohl logische als auch physikalische Verbindungen werden erst nach Bestätigung der Verbindungsabbaumeldung durch die Anwendung (`DISCONNECT_RESP/_B3_RESP`) ungültig. Erst dann können NCCI bzw. PLCI erneut von der CAPI-Instanz vergeben werden.

Abbau der logischen Verbindung

Bestätigt die Anwendung den Abbau einer logischen Verbindung (`DISCONNECT_B3_RESP`) vom *session*-Typ `Secure`, verliert der ausgehandelte *session key* seine Gültigkeit. Er wird, um jeden späteren Ausleseversuch zu entmutigen, vom Modul *security control* aus der SIB entfernt. Anschließend wird der NCCI aus der SIB ausgetragen, bevor die Meldung an die CAPI-Instanz weitergereicht wird:

```
void erase_session_key (unsigned long ncci);
void remove_ncci (unsigned long ncci);
```

Abbau der physischen Verbindung

Wird der erfolgreiche Abbau einer physikalischen Verbindung vom *connection*-Typ `Secure` von der Anwendung bestätigt (`DISCONNECT_RESP`), löscht das Modul *communication control* den zugehörigen PLCI aus der SIB, bevor die Meldung an die CAPI-Instanz weitergegeben wird:

```
void remove_plci (unsigned long plci);
```

5. FAZIT

Es wurde die Spezifikation eines auf dem CAPI aufsetzenden Sicherheitstreibers S-CAPI vorgestellt, der die Sicherheitsdienste Modifikationserkennung, Authentizität und Vertraulichkeit realisiert. Die Dienste werden, unabhängig sowohl von den darüberliegenden Kommunikationsanwendungen als auch von der eingesetzten ISDN-Hardware, vollständig

transparent integriert. Die Spezifikation abstrahiert weitgehend von kryptographischen Verfahren und Protokollen; es erfolgte lediglich die Festlegung auf symmetrische Verschlüsselungsverfahren und asymmetrische, zertifikatbasierte Authentikations- und Schlüsselaustauschprotokolle. Die Mechanismen wurden in einzelnen Modulen gekapselt und können ihrerseits auf Krypto-APIs aufsetzen.

Das *Common ISDN API* (CAPI) eignet sich besonders gut für die Integration der genannten Sicherheitsdienste: Es ist inzwischen Basis der meisten PC-basierten ISDN-Kommunikationslösungen und abstrahiert zudem so weitgehend von ISDN-spezifischen Eigenschaften, daß Konzeption und Spezifikation des S-CAPI in weiten Teilen auf andere Kommunikationsumgebungen (z.B. LANs) übertragen werden können.

DANK

Detlef Dienst, Olaf Junklewitz, Rudi Schöngarth und Dieter Schmidt realisierten im Rahmen von Studien- und Diplomarbeiten einen S-CAPI-Prototyp für DOS- und MS-Windows-basierte ISDN-Anwendungen. Die Firmen CE Infosys GmbH und KryptoKom GmbH unterstützten das Projekt durch die Überlassung von DES-Hardware resp. eine schnelle DES-Softwareimplementierung. Gudula Kiehle danken wir für ihre gründliche Durchsicht dieses Beitrags.

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 CAPI-Arbeitskreises der Telekom, Projekt ROLAND, Version 2.0, Februar 1994.

3. Davies, Donald W.; Price, Wyn L.: "Security for Computer Networks". 2. Auflage, John Wiley & Sons Ltd., Chichester 1989.

4. Diffie, Whitfield; Oorschot, Paul C. van; Wiener, Michael J.: "Authentication and Authenticated Key Exchanges". *Designs, Codes & Cryptography*, No. 2/92, S. 107-125.

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

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

7. Horster, Patrick; Knobloch, Hans-Joachim: "Protokolle zum Austausch authentischer Schlüssel". In: Verläßliche Informationssysteme. Proceedings der Fachtagung VIS '91, *Informatik Fachberichte* Nr. 271, Springer, Heidelberg 1991, S. 321-328.

8. International Organisation for Standardization (ISO): "Open Systems Interconnection: Basic Reference Model". International Standard ISO 7498, 1983.

9. 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.

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

11. 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.

12. Needham, Roger M.; Schroeder, Michael D.: "Using Encryption for Authentication in Large Networks of Computers". *Communications of the ACM*, Bd. 21, Nr. 12/78, S. 993-999.

13. Ruland, Christoph: "Informationssicherheit in Datennetzen". DataCom-Verlag, Bergheim 1993.