Realisierung einer Verschlüsselungstechnik für Daten im ISDN B-Kanal

Diplomarbeit

Technische Fachhochschule Berlin
Fachbereich 13 - Technische Informatik
Wintersemester 1997/1998

Boris Floricic

Betreuer:	Prof. Dr. rer. nat. C. Kordecki
Gutachter:	Prof. Dr. Ing. Buchholz
Vorsitzender:	Prof. Bormann

Diplomarbeit
"Realisierung einer Verschlüsselungstechnik für Daten im ISDN B-Kanal"

Inhaltsverzeichnis

1 Einleitung

In diesem Kapitel wird der Aufbau der vorliegenden Diplomarbeit erklärt. Die Arbeit besteht aus 6 Kapiteln plus Anhang. Die Einleitung, das erste Kapitel, lesen Sie gerade. Im zweiten Kapitel wird die Motivation und Zielstellung beschrieben. Warum wird das beschriebene Gerät gebraucht und welche Eigenschaften soll es erfüllen. Das dritte Kapitel befaßt sich mit dem Stand der Technik und dem technischen Umfeld. Wie ist der ISDN-Basisanschluß aufgebaut und welche Eigenschaften des ISDN-Netzes sind für diese Arbeit interressant? Und es werden die Grundlagen der Kryptographie betrachtet. Zudem wird in diesem Kapitel die Auswahl der relevanten Bauelemente für das Projekt getroffen. Im vierten Kapitel wird der Kryptoalgorithmus, der für dieses Projekt ausgewählt worden ist, beschrieben. Es wird ein Überblick über die gesamte Hardware des Verschlüsselungstelefons gegeben. Das fünfte Kapitel geht dann ins Detail. Die Hardwarekomponenten und auch die Implementierung der Kroyptoalgorithmen werden dort näher beschrieben. Als letztes Kapitel folgt eine abschließende Zusammenfassung der Arbeit und eine eigene Bewertung. Es wird auch beschrieben, was und wie dieses Gerät noch verbessert und weiterentwickelt werden kann.

Im Anhang befinden sich Tabellen und Diagramme, Literatur- und Abbildungsverzeichnisse, Listen der Signalnamen und deren Bedeutung, die Pinbelegung der Steckleisten, eine Stückliste der eingesetzten Bauelemente und einiges mehr. Zudem enthält der Anhang die Stromlaufpläne, Layouts und Bestückungspläne. Dieser Diplomarbeit ist zudem eine Daten-CD beigelegt, deren Inhaltsangabe ebenfalls dem Anhang zu entnehmen ist.

2 Motivation und Zielsetzung

In dieser Diplomarbeit wird der Aufbau eines Sprachverschlüsselungssystems behandelt, das am ISDN-Basisanschluß betrieben werden kann. Es gewährleistet weitgehen die Geheimhaltung der Kommunikation zwischen zwei Gesprächspartnern. Die Privatsphäre wird zur Zeit immer mehr gestört oder eingeschränkt. Angefangen von Hackern, die aus Spaß an der Sache fremde Leitungen anzapfen und mithören, über professionellen Datendiebstahl, z.B. Industriespionage, bis hin zu staatlichen Maßnahmen wie "Der große Lauschangriff" reicht die Palette der Angriffe. Man sollte sich seine Privatsspäre sichern und sich nicht zum "gläsernen Menschen" machen lassen.

Ziel ist es, ein Gerät zu entwickeln, das die zu übertragenden Informationen zwischen den Endgeräten so unkenntlich für Dritte macht, daß nur die beiden Gesprächspartner ihre Nachrichten verstehen können. Die Daten, die über den hausinternen ISDN-S0-Bus, den UK0-Leitungen zu den Vermittlungsstellen und durch die Vermittlungsstellen gehen, werden verschlüsselt und können nur mit dem entsprechenden Code wieder entschlüsselt werden.

Der Grund, daß für dieses Projekt das ISDN-Netz und nicht das analoge Telefonnetz genutzt wird, besteht darin, daß bei ISDN die Sprachdaten schon in digitaler Form vorliegen und nicht erst komprimiert und auf die analogen Leitungen aufmoduliert werden müssen.

Es soll also ein komplettes ISDN-Endgerät, entsprechend der Funktionalität eines Telefons, aufgebaut werden, das noch zusätzlich die Nutzdaten im jeweiligen B-Kanal ver- und entschlüsselt. Die verwendbaren Verschlüsselungsalgorithmen sollten einen hohen Sicherheitsstandard aufweisen. In der heutigen Zeit ist es möglich, mit entsprechendem matriellen Einsatz, wie ihn sich einige größere Organisationen oder staatliche Behörden leisten können, einfachere Codierungen zu entschlüsseln. Der Schlüssel sollte, wie bei einem realen Schloß, entfernbar und austauschbar sein. Dafür wird in dieser Arbeit, ähnlich einer Telefonkarte, eine Chipkarte eingesetzt, die den Schlüssel enthält. Besser währen Karten, die sowohl den Schlüssel, als auch einen Teil des Algorithmuses unterbringen, wie z.B. die NetKeyCard von Telesec.

Für ein ISDN-Endgerät muß das D-Kanal Protokoll (Siehe Seite 10) auf einem Steuerprozessor implementiert werden. Dieses D-Kanal Protokoll ist nicht Teil der Diplomarbeit, sondern wird in einer getrennten Arbeit behandelt.

Hier geht es vielmehr um die gesamte Hardware und die Verschlüsselungsalgorithmen. Es soll der Aufbau eines ISDN-Telefons mit Verschlüsselungsfunktion gezeigt werden. Softwareseitig wird auf Ver- und Entschlüsselung von Datenströmen eingegangen.

Eine besondere Anforderung an die Hardware ist, daß sie aus Bauteilen besteht, die leicht im Elektronikhandel erhältlich sind. Denn diese Schaltung soll von jedermann leicht nachbaubar sein. Es sollen also keine exotischen Spezialbauteile und keine Chips, die mit speziellen Programmiergeräten gebrannt werden müssen, eingesetzt werden. Die Leiterplatte(n) sollen maximal zweiseitig sein, damit sie noch relativ einfach zu fertigen sind. Auch dürfen die verwendeten Bauteile nicht ein zu kleines Pinraster haben, damit sie auch von jedem leicht zu verlöten sind. Also kommen erstmal nur DIL und PLCC Gehäuse in Frage.

3 Stand der Technik

Hier wird die Basis für dieses Projekt beschrieben. Als erstes, was es schon an vergleichbaren Geräten gibt und was sonst noch so geplant ist. Als zweites, für die Arbeit alle wesentlichen Fakten über den ISDN-Basisanschluß. Als letztes noch, welche Halbleiter für dieses Gerät in Frage kommen.

3.1 Was schon an Vergleichbarem existiert

Es existieren schon Einrichtungen, mit denen verschlüsselt auf dem ISDN-Netz telefoniert werden kann. Die meistgenutzte Variante ist der Einsatz eines Personalcomputers mit einer ISDN-Karte und einer Soundkarte. Die ISDN-Karte im PC arbeitet mit einem hardwarespezifischen CAPI-Treiber zusammen. Dieser CAPI-Treiber übernimmt alle Hardwarefunktionen der Karte und auch die gesamte Verwaltung des D-Kanals. Nun gibt es Programme, meist unter Windows, die mit Hilfe einer Soundkarte ein ISDN-Telefon simulieren. Ein Kopfhörer mit Mikrophon wird an die Soundkarte angeschlossen und damit ein Telefonhörer geschaffen. Auf einem Fenster der graphischen Oberfläche werden dann Tastenfeld und Display dargestellt. Nun kann noch bei einigen Programmen eine Verschlüsselungsfunktion aktiviert werden, um Gespräche mit einer gewissen Geheimhaltung zu führen. Die Ver- und Entschlüsselung wird dann vom Hauptprozessor des PC durchgeführt. Der Sinn dieser Variante ist, wenn schon ein PC mit ISDN und Soundkarte zur Verfügung steht, daß nur noch diese Software installiert werden muß, um verschlüsselt auf dem ISDN-Netz telefonieren zu können. Der Nachteil ist aber, daß für jedes Telefongespräch extra der PC mit Windows hochgefahren werden muß. Desweiteren ist ein kompletter PC nur zum Telefonieren z.zt. noch sehr aufwendig. Besser wäre ein Gerät wie ein normales Telefon, das eine Verschlüsselungsfunktion schon mit eingebaut hat.

Die Firma Siemens baut unter dem Namen "DSM ISDN" ein solches Gerät, aber nähere Informationen sind derzeit nicht erhältlich. Das Verschlüsselungstelefon ist nicht für die Allgemeinheit bestimmt. Auch von der Telekom (Telesec) werden in einiger Zeit verschiedene Dienste gegen das illegale Abhören von Fernmeldeleitungen angeboten, die jedoch nicht generell vor einem Abhören schützen.

Für das analoge Telefonnetz hat der Chaos-Computer-Club schon ein "Crypto-Phone" entwickelt. Es benutzt ein Modem zur Verbindung mit dem Telefonnetz und hat eine GSM-ähnliche Sprachkompression. Hauptbestandteil ist ein leistungsstarker DSP. Das Projekt ist neu, und es sind nur wenige Informationen zu bekommen.

3.2 ISDN-Basisanschluß

Jeder normale Telefonanschluß kann, auf Antrag, in einen ISDN-Basisanschluß umgewandelt werden, wenn eine Angliederung an eine digitale Vermittlungsstelle zur Verfügung steht. Ob ein Anschluß an einer digitalen Vermittlungsstelle vorhanden ist, kann ganz einfach festgestellt werden durch den Versuch, mit Ton-Wahl (DTMF) zu wählen. Ist nur der Puls-Wahl-Modus möglich, dann ist man noch an einer alten analogen Vermittlungsstelle angeschlossen, und es wird etwas länger dauern, bis der ISDN-Anschluß geschaltet wird. Ein ISDN-Anschluß ist z.zt. teurer als ein normaler Telefonanschluß, er hat aber auch einige Vorteile. Einer der zwei wichtigsten ist, daß auf einem einzigen Leitungspaar, der UK0-Leitung, gleichzeitig zwei Kommunikatiosverbindungen nach draußen geführt werden können, was z.B. eine Dreierkonferenz ermöglicht, weiterhin Sprechen und Faxen gleichzeitig, Angerufenwerden während telefoniert wird und vieles mehr [2]. Der andere große Vorteil ist die viel höhere Übertragungsrate von Daten. Das schnellste verfügbare Modem kann auf einer normalen Telefonleitung 33,6kBit/s (spezielle Modems 56,7kBit/s) übertragen. ISDN kann 64kBit/s, oder wenn beide Kanäle benutzt werden, sogar 128kBit/s (kostet aber auch doppelt Gebühren). Bei ISDN-Anschlußverlegung wird von der Telekom ein NT-BA an die Leitung angeschlossen, an der sich zuvor das analoge Telefon befunden hat (Abbildung 1). Damit wird die Leitung zur Vermittlungsstelle zur Uk0-Schnittstelle, an der natürlich kein analoges Telefon mehr angeschlossen werden darf, da sich die elektrischen Eigenschaften geändert haben.

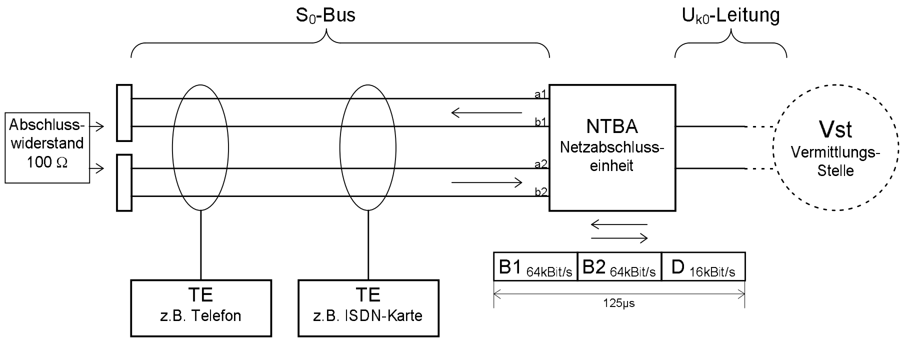
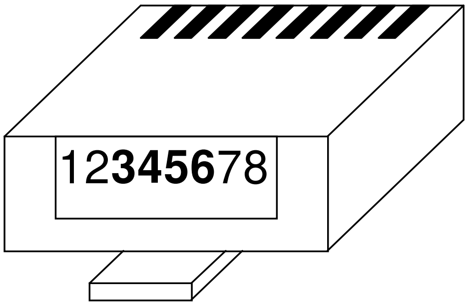


Abbildung 1: ISDN-Basisanschluß

3.2.1 ISDN S0-Bus

An dem NTBA steht der S0-Bus zur Verfügung. Dort können nun die ISDN-Endgeräte (Telefon, ISDN-Karte, u.s.w.) oder eine ISDN-Telefonanlage angeschlossen werden (Abbildung 1). An diesem Bus können max. acht Geräte betrieben werden, die mit maximal zehn Meter langen Stichleitungen am Bus angeschlossen werden. Der gesamte S0-Bus darf 150 Meter bei Mehrgeräteanschluß oder 1000 Meter bei einem einzigen angeschlossenen Gerät lang sein. Dieser S0-Bus ist vieradrig, davon sind zwei Adern für den Transport der Daten vom Endgerät (TE) zum Netzabschlußadapter (NTBA oder NT) belegt und zwei für die Daten vom NT zum TE. Eine Stromversorgung für die Endgeräte wird zusätzlich mit auf dem S0-Bus zur Verfügung gestellt (Abbildung 2). Es handelt sich hierbei um eine Spannung von 40V zwischen den Leitungspaaren a1-b1 und a2-b2, die mit max 100mA belastet werden darf. Die Signale werden mittels eines Übertragers in den ISDN-Geräten aus den Leitungspaaren ausgekoppelt.

Beim S0-Bus repräsentiert der Netzabschlußadapter (NTBA oder NT) die Vermittlungsstelle, weil vom NTBA die Uk0-Leitung zur Vermittelungsstelle geht. Der S0-Bus sieht nur den NTBA und hat eine Übertragungsrate von 144 kBit/s netto pro Richtung. Das sind 2 B-Kanäle mit je 64 kBit/s und ein D-Kanal mit 16 kBit/s. Es gibt zudem noch Synchronisations- und Steuerbits (Abbildung 3) auf dem Bus, die hier nicht weiter betrachtet werden, da sie für die Verschlüsselung der Nutzdatenkanäle nicht weiter wichtig sind und diese Bits schon automatisch in allen am Markt verfügbaren ISDN-Chip abgearbeitet werden. Für die Verschlüsselung sind eigentlich nur die Nutzdatenkanäle, die B-Kanäle, interessant.



3	b2	TE → NT	neg. Puls	pos. Versorgungsspannung
4	b1	NT → TE	neg. Puls	neg. Versorgungsspannung
5	a1	NT → TE	pos. Puls	neg. Versorgungsspannung
6	a2	TE → TE	pos. Puls	pos. Versorgungsspannung

Abbildung 2: ISDN Western-Stecker

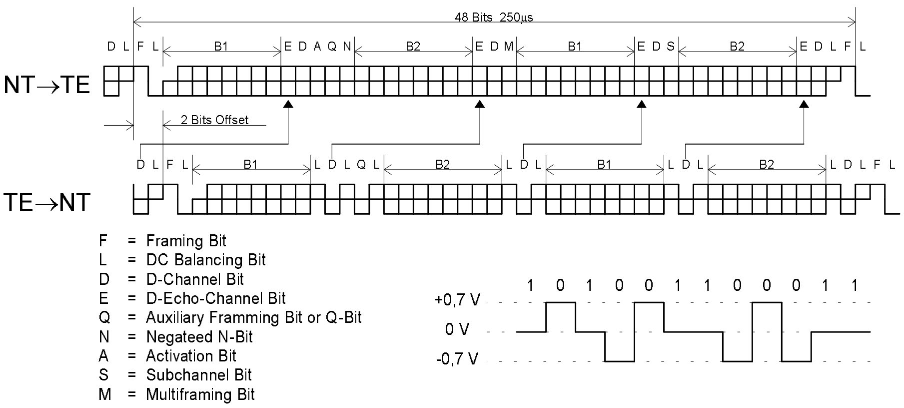


Abbildung 3: Signale auf dem ISDN S0-Bu

3.2.2 ISDN D-Kanal

In diesem Signalisierungskanal, der wichtigsten Einheit im ISDN-System, werden sämtliche Steuerfunktionen übertragen. Die ganze Organisation wie z.B. Verbindungsauf- und -abbau, ankommender Ruf (Klingel), gewählte Ziffern, Rufnummer des Gesprächspartners, Gebühreninformationen und vieles mehr wird auf dem D-Kanal übertragen. Als wichtig sei hier erwähnt, daß im D-Kanal anzugeben ist, welcher Art die Nutzdaten auf den B-Kanälen sind (Sprache oder Daten). Das ist insofern für die vorliegende Diplomarbeit wichtig, da ein Telefon "restricted" Sprechdaten anmeldet, aber diese nach dem Verschlüsseln nur noch "unrestricted" Binärdaten sind.

Das sehr komplexe ISDN-D-Kanal Protokoll ist in 3 Schichten (Abbildung 4) aufgeteilt. Die erste Schicht ist die Hardware- und Bitübertragungsschicht. Die Richtlinien für die Schicht 1 des D-Kanals sind in der ITU-T I.430 / I.431 Norm festgelegt. Beim S0-Bus ist eine 0 dominant und die 1 rezessiv. Diese Festlegung ist wichtig, da an dem Bus bis zu 8 Geräte gleichzeitig angeschlossen sein können, die ein D-Kanal Paket senden wollen. Daher gibt es eine Kollisionserkennung, die dem Gerät mit der

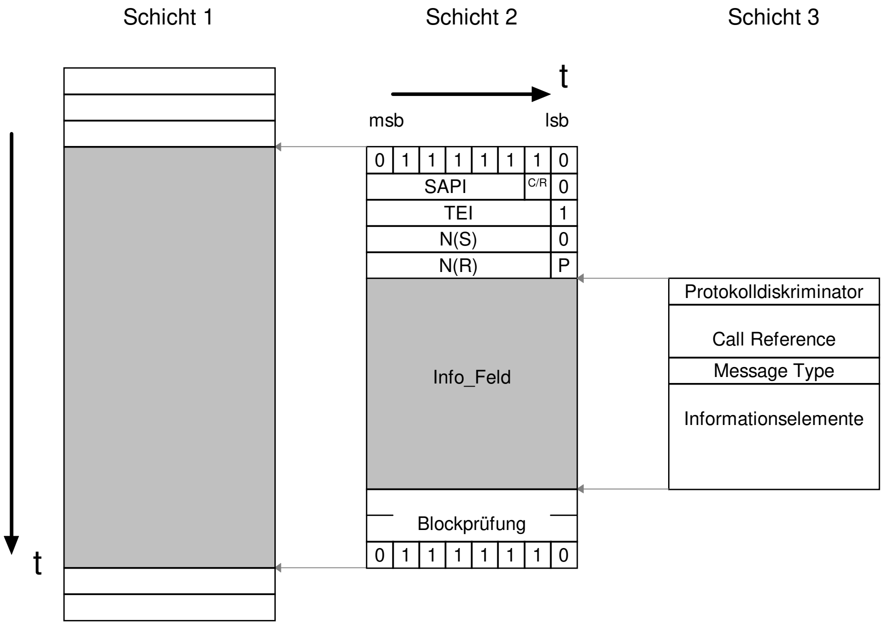


Abbildung 4: D-Kanal Schichtenmodel

niedrigeren Priorität veranlaßt, die Sendung einzustellen. Da die D- Kanal Informationen bitweise übertragen werden, muß eine Bytesynchronisation stattfinden. Diese besteht aus der Bitfolge "01111110" und wird immer am Anfang und Ende eines D-Kanal Pakets übertragen. Damit es keine Verwechslungen mit Daten gibt, die auch zufällig sechs oder mehr Einsen hintereinander haben, wird nach dem "Bitstuffing" Algorithmus immer nach fünf aufeinanderfolgenden Einsen vom Sender eine Null eingefügt, die vom Empfänger wieder entfernt wird. Damit ist sichergestellt, daß die Bitfolge "01111110" nur für die Bytesychronisation zuständig ist, und acht oder mehr Einsen kennzeichnen einen freien D-Kanal. Bei der Bitfolge "011111110" wird ein Rahmenabbruch durchgeführt. Dieses Verfahren stammt aus dem HDLC-Protokoll, welches für X.25 eingesetzt wird.

In der zweiten Schicht liegt die Aufgabe der Fehlererkennung und -behandlung sowie die Adressierung der Pakete auf dem D-Kanal, damit diese das richtige Gerät erreichen. Die ITU-T Norm für Schicht 2 ist Q.920 / Q.921.

Die dritte Schicht enthält nun die eigentlichen Signalisierungsdaten. Es gibt zur Zeit zwei Standards für diese Schicht, die von der Telekom in Deutschland eingesetzt werden. Die eine ist die alte nationale 1TR6 Norm, die von der Bundespost eingeführt wurde. Sie wird noch voraussichtlich bis ins Jahr 2003 von der Telekom unterstützt. Die andere ist die europaweit geltende Norm DSS1 ITU-T Q.930 / Q.932. Diese Norm ist die allgemein gültige für einen zukünftigen Basisanschluß.

3.2.3 ISDN B-Kanal

Sobald mit Hilfe des D-Kanals eine Verbindung zwischen zwei Partnern aufgebaut worden ist, werden die eigentlichen Nutzinformationen auf dem B-Kanal übertragen. Welcher der beiden B-Kanäle für diese Verbindung benutzt wird, wurde zuvor mit dem

D-Kanal signalisiert.

Die Informationen auf dem B-Kanal werden byteweise übertragen, und im Rahmen dieser Arbeit entfällt die Bytesynchronisation. Es werden pro B-Kanal 64 KBit/s oder dementsprechend 8 KByte/s übertragen. Im B-Kanal gibt es keine weitere Schichtenaufteilung, da diese Bytes gleich von der Anwendungsschicht genutzt werden. Beim ISDN-Telefon sind diese Bytes zugleich die Werte für den A/D und D/A-Wandler. Die 8 Bits werden lediglich auf 14 Bit mit dem A-law Verfahren expandiert. Diese Bit-Expansion und Kompression ist in der IUT-T G.711 beschrieben. Durch ISDN-Karten werden die Bytes aus dem ISDN-Kanal direkt in den Computer übertragen und dort nach Bedarf weiterverarbeitet.

Für die Verschlüsselung der B-Kanal Daten ist folgendes zu beachten:

Der konstante Bytestrom von 8000 Byte/s kann fehlerhafte Bytes enthalten. Es muß nun dafür gesorgt werden, daß dieser Bytestrom beim Sender verschlüsselt und beim Empfänger entschlüsselt wird und sich somit wieder die ursprünglichen Daten ergeben, obwohl durchaus ein Byte während der Übertragung gestört werden kann. Welcher Art die eigentlichen Nutzdaten sind (Sprache, Fax oder Daten) ist unerheblich. Der Vermittlungsstelle muß lediglich über dem D-Kanal mitgeteilt werden, daß "unrestricted" Binärdaten übertragen werden, damit diese nicht als Sprachdaten interpretiert werden und unter Umständen z.B. durch GSM-Sprach- Compression oder analoge Leitungen einer Verfremdung unterliegen.

3.3 Verschlüsselungsalgorithmen

Ver- und Entschlüsselungsalgorithmen sind besondere Rechenvorschriften mit zwei Eingangs- und einer Ausgangsvariablen [1]. Einer der Eingangswerte ist immer der Schlüssel (k). Für die Verschlüsselung (E) ist der andere Eingangswert der Klartext (m), und der Ausgangswert ist der Chiffretext (c). Beim Entschlüsseln (E-1) ist es umgekehrt, der Chiffretext (c) geht in den Algorithmus rein, und es kommt bzw. sollte herauskommen der Klartext (m), falls der Schlüssel (k) stimmt. Das besondere an diesen Algorithmen ist folgendes: Es dürfen keinerlei Rückschlüsse auf den Schlüssel (k) bzw. den Inhalt des Klartextes (m) gezogen werden können, wenn nur der Chiffretext (c) bekannt ist. Ebenfalls darf auch der Schlüssel (k) nicht mit der Kenntnis von Chiffre- und Klartextpaaren zurückgerechnet werden können. Die Sicherheit eines Verschlüsselungsalgorithmusses hängt von der Länge des Schlüssels (k) ab. Theoretisch ist es bei symmetrischen Algorithmen so, daß mit jedem Bit um das der Schlüssel (k) länger wird, es doppelt so schwer wird, ihn durch durch Austesten offenzulegen. Es gibt nun mehrere Arten von Verschlüsselungsalgorithmen, die in zwei Klassen von Algorithmen aufgeteilt werden können:

- Symmetrische Verschlüsselungsalgorithmen
  Blockchiffrierer, Stromchiffrierer, Oneway-Algorithmen

- Asymetrische Verschlüsselungsalgorithmen
  RSA, Diffie Hellman Schlüsselaustausch

3.3.1 Blockchiffrierer

Die Blockchiffrierer verschlüsseln einen Klartextblock mit konstanter Größe, meist 64 Bit, zu einem Chiffretextblock mit gleicher Größe, und beim Entschlüsseln entsprechend umgekehrt. Es wird beim Ver- und Endschlüsseln der gleiche Schlüssel (k) verwendet. Daher sind diese Algorithmen symmetrisch. Solche Algorithmen bestehen meist aus Iterationen von Bitpermutation, Shiften, XOR, Lookuptables und besonderen Multiplikationen. Die bekanntesten Algorithmen sind DES und IDEA.

3.3.2 Stromchiffrierer

Diese Chiffrierer sind ebenfalls symmetrisch, aber es gibt keine konstante Größe der Klar- und Chiffretextblöcke. Sie ver- und entschlüsseln Bit- oder Byteströme. Je nach Art des Algorithmusses sind die Verschlüsselungsmuster nur vom Schlüssel (k) oder vom Schlüssel (k) zusätzlich des Datenstromes abhängig. Entweder handelt es sich bei ihnen um reine Stromchiffrierer wie RC4 oder A5-GSM, oder es werden Blockchiffrierer mit besonderen Rahmenalgorithmen eingesetzt und so zu Stromchiffrierern umfunktioniert.

3.3.3 Asymmetrischer RSA Algorithmus

Besonders interessant sind die asymmetrischen Algorithmen, wie RSA, in denen für die Ver- und Entschlüsselung zwei verschiedene Schlüssel (kS, kP) eingesetzt werden. Der Schlüssel (kS) bleibt geheim, und der Schlüssel (kp) wird veröffentlicht. Das hat den großen Vorteil, daß jeder eine Nachricht mit dem öffentlich verfügbaren Schlüssel (kP) verschlüsseln kann, aber nur mit dem privaten Schlüssel (kS) diese Nachricht wieder entschlüsseln kann. Der Algorithmus hat die recht einfache aber doch sehr wirkungsvolle Formel: c = mk MOD n" (m=Klartext, c=Chiffretext, n=Produkt zweier Primzahlen und zusammen mit k ein Teil des Schlüssels). Das Geheimnis liegt in der Auswahl der Zahlen kS, kP und n. Die Zahlen kS, kP und n müssen sehr groß (1024 Bit oder mehr) gewählt werden, damit der Algorithmus sicher ist. Dieser Algorithmus eignet sich aber nicht, um direkt Daten zu verschlüsseln, sondern es muß zusätzlich ein symmetrischer Algorithmus hinzugenommen werden, um die eigentlichen Datenströme zu verschlüsseln (Hybritalgorithmus) (Abbildung 5). Genau das macht das sehr gute Programm PGP, das für die Verschlüsselung von eMails im Internet verwendet wird [5]. Es verwendet eine Kombination von RSA und IDEA.

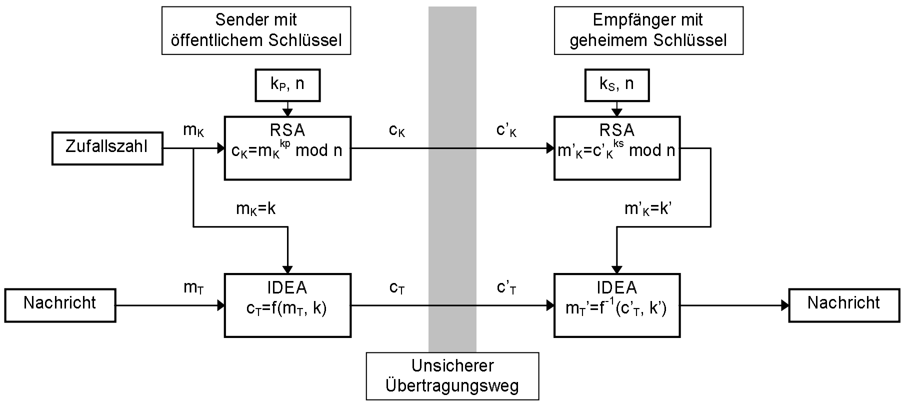


Abbildung 5: Hybrid-Algorithmus mit RSA und IDEA (PGP)

3.3.4 Diffie Hellman Schlüsselaustausch

Noch eine elegante Art, den symmetrischen Schlüssel auszutauschen, ist der Diffie Hellman Schlüsselaustausch. Hier können sich zwei Partner, die zuvor noch keine Daten ausgetauscht haben, in aller Öffentlichkeit auf einen geheimen Schlüssel für einen symmetrischen Algorithmus einigen, ohne das Dritte ihn erfahren. Genauso wie beim RSA basiert alles auf der Rechnung "c = mk MOD n", und es müssen auch hier sehr große Zahlen (1024Bit oder mehr) gewählt werden.

Partner A	Öffentlichkeit	Partner B
	p (Primzahl)	
	$s(N und\, s < p)$	
$a(N und\, a < p)$		$b(N und\, b < p)$
$\alpha = s^a \bmod p$		$\beta = s^b \bmod p$
	α, β	
$k = \beta^a \bmod p$		$k = \alpha^b \bmod p$

Abbildung 6: Ablauf des Diffie Hellman Schlüsselaustauschs

Für den Schlüsselaustausch (Abbildung 6) einigen sich beide Partner in aller Öffentlichkeit auf eine Primzahl (p) und eine natürliche Zahl (s) die kleiner als p ist. Jetzt sucht jeder Partner für sich je eine natürliche Zahl (a, b) aus, die auch kleiner als p sein muß, und jeder errechnet sich damit eine Zahl (α, β), die sich die Partner in aller Öffentlichkeit austauschen. Die Zahlen a und b bleiben geheim. Nun kann sich jeder Partner die Zahl k errechnen. Ein Dritter, der die Zahlen p, s, α und β mitgehört hat, kann nicht die Zahl k berechnen, weil er weder die Zahl a noch die Zahl b kennt. Die beiden Partner können aber die Zahl k nun als Basis für einen symmetrischen Algorithmus nutzen. Die große Gefahr bei diesem Schlüsselaustausch ist, daß sich ein Dritter die Leitung auftrennt und dann nach beiden Partnern hin getrennt je einen Schlüsselaustausch vornimmt. Beide Partner denken, sie hätten einen gemeinsamen geheimen Schlüssel k, aber dabei haben sie jeweils mit dem Angreifer den Schlüsselaustausch vorgenommen und sind auch im Besitz verschiedener Schlüssel k1 und k2. Dieses Verfahren erfordert es unbedingt, mit seinem Partner die Checksumme des Schlüssels k zu vergleichen, und zwar so, daß es der Angreifer nicht verfälschen kann.

3.3.5 "Oneway" Algorithmen

Dann wären da noch die Oneway-Algorithmen zu erwähnen, die, wie der Name schon sagt, nur verschlüsseln können. Für sie ist auch meist der erzeugte Chiffretext (c) kürzer als der eingegebene Klartext (m). Diese Algorithmen machen durchaus Sinn, besonders für die Authentifizierung beispielsweise in der Passwortverwaltung. Nach der Eingabe des Passwortes auf der Tastatur wird das Passwort verschlüsselt und mit dem verschlüsselten Passwort in einer Datenbank verglichen. Das Passwort kann nicht mehr zurückgerechnet werden, es kann aber immer noch mit dem eingegebenen verglichen werden. Oneway-Algorithmen sind meist einfacher, da keine Invertierung erforderlich ist. Meist sind sie anwenderspezifisch gestaltet oder es wird die Verschlüsselungsfunktion von Standard-Algorithmen genutzt. Ihren Einsatz finden diese Algorithmen oft bei Chipkarten zur Authentifizierung, z.B. Pay- TV, Mobiltelefone oder Zugangskontrolle. (Abbildung 7)

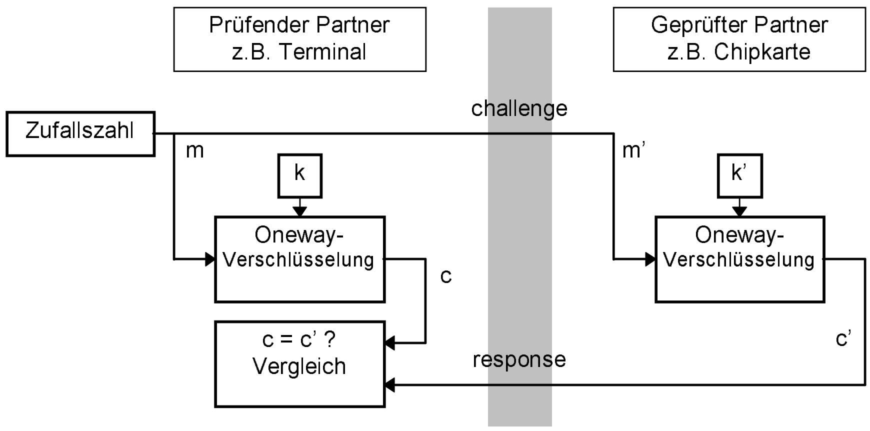


Abbildung 7: Oneway-Algorithmus zur Authentifizierung

3.4 Bausteine für dieses Projekt

Von besonderer Bedeutung ist neben der eigentlichen Funktionalität die Auswahl der Bausteine, damit das fertige Gerät auch bestmöglich die in der Aufgabenstellung genannten Eigenschaften erfüllt: leicht nachzubauen, hohe und preiswerte Verfügbarkeit der Chips und allgemeinen Bauelemente.

3.4.1 ISDN-Bausteine

Der ISDN-Chip stellt insofern ein Problem dar, da er mehr zu den Spezialbauteilen gehört und damit schwer beschaffbar ist. Einer der größten Anbieter für ISDN-Bausteine ist die Firma Siemens. Die Bausteine, die hier in Frage kämen, wären PEB2080, PEB2085/6 und PEB2185/6. Auch andre Firmen bieten ISDN-Chips an, wie National Semiconductor den TP3420N, Mintel den MT9830 oder Motorola den MC145572. Aber AMD hat mit dem AM79C30A etwas Besonderes auf den Markt gebracht [7]. Er enthält als einziger Chip jegliche Funktionalität, die für die ISDN-Anbindung derzeit nötig ist, angefangen vom S0-Bus-Interface bis hin zu dem CODEC mit seinem A/D, D/A Wandler mit Filter-DSP. Außerdem nimmt er einem auch eine Menge Arbeit in den unteren Schichten des D-Kanalprotokolls ab. Desweiteren verfügt er über eine serielle Schnittstelle für Nutzdatenkanäle, die sich gut mit einem DSP verbinden läßt. Auch ist er nicht viel schwerer beschaffbar als die alternativ zuvor erwähnten ISDN-Bausteine und befindet sich in einem PLCC-Gehäuse.

3.4.2 Prozessoren für die Verschlüsselung

Der verwendete Prozessor oder die verwendeten Prozessoren müssen zwei Kriterien erfüllen. Zum einen muß er schnell genug sein, um je einen 8kByte/s großen Datenstrom mit IDEA zu ver- und entschlüsseln. Und zum anderen muß er leicht beschaffbar sein und sollte auch nicht viel kosten. Da der IDEA-Algorithmus im Cipher-Feedback-Mode genutzt werden soll und es ja zwei Datenströmme sind, einer hin und einer zurück, müssen hier 16000 IDEA-Berechnungen pro Sekunde bewerkstelligt werden. Der IDEA-Algorithmus besteht hauptsächlich aus Multiplikationen. Daher bieten sich DSPs an, da sie Multiplikationen sehr viel schneller ausführen als normale Prozessoren. Aber die in die engere Wahl gezogenen DSPs eignen sich aufgrund ihres Befehlssatzes schlecht für Steueraufgaben, sondern eher für Berechnungen, die in der Regelungstechnik anfallen. Eine andere gute Lösung wäre ein komplettes, fertiges Prozessorboard wie das Intel i386EX. Es hat einen 386er mit 25MHz und je 1MByte Flash-EEPROM und RAM. Und es ist schon fertig aufgebaut, nur die ISDN-Hardware, Display und Tastatur müssen noch angefügt werden. Aber selbst der 386er mit 25MHz ist noch ca. 3x zu langsam für die 16000 IDEAs pro Sekunde, und er ist mit ca. 600 DM auch sehr teuer. Also zurück zu den DSPs. Texas Instruments hat die DSP-Serie TMS320Cxx. Dort gibt es einige DSPs, die über internen RAM-Speicher und Bootloader verfügen. Wenn dieser RAM-Speicher ausreicht, um das IDEA-Programm und seine Daten zu verwalten, wird kein weiterer externer Speicher mehr für die DSPs benötigt. Sie müßten dann aber von einem weiteren Prozessor nach jedem Einschalten mit ihrem Programm geladen werden. Aber ein zusätzlicher Prozessor wird ohnehin benötigt, da sich DSPs, wie schon erwähnt, nicht gut für die Bedienung der Steuersignale des D-Kanals und des Benutzerinterfaces eignen. Der TMS320C26 ist eine gute Wahl, da er die Kriterien der Aufgabenstellung erfüllt. Er ist leicht beschaffbar, preiswert (ca. 33,- DM), besitzt ein PLCC-Gehäuse und benötigt fast keine externen Bauteile [6]. Jedoch schafft der TMS320C26 mit 40MHz nur ca. 11000 IDEAs pro Sekunde. Da aber mit zwei separaten Datenströmen von je 8000 Byte/s gearbeitet wird, kann mittels zweier DSPs einerseits der Datenstrom vom Benutzer zur Vermittlungsstelle verschlüsselt und mit dem anderen DSP der Datenstrom von der Vermittlungsstelle zum Benutzer entschlüsselt werden. Ein weiterer positiver Aspekt ist diesbezüglich die serielle Schnittstelle, über die dieser DSP verfügt. Sie läßt sich sehr gut mit den Datenströmen, die aus dem ISDN-Chip kommen, verschalten. Als Steuerprozessor kommt hier ein schnelles 51er-Derivat von Dallas zum Einsatz, das ein paar wesentliche Vorteile gegenüber dem normalen 80C51 hat. Die wichtigsten bei diesem Bauteil genutzten Vorteile sind, daß es die Befehle im Durchschnitt 2,5 mal schneller ausführt als ein normaler 51er Prozessor bei gleicher Taktfrequenz, die hier sogar über 33MHz liegen kann. Ein weiterer Vorteil ist, daß es über einen zweiten seriellen Port verfügt. Desweiteren hat der Dallas Prozessor zwei Datenpointer, die zur Adressierung der externen Datenspeicher genutzt werden können.

4 Grobkonzept

Hier werden die beiden wichtigsten Aspekte dieses Projekts vorerst grob umrissen. Dazu gehören die eingesetzten Verschlüsselungs-Algorithmen sowie die einzusetzende Hardware.

4.1 Verschlüsselungsalgorithmus für den Datenstrom

Da es sich beim B-Kanal um einen Bytestrom handelt, der auch fehlerhafte Bytes enthalten kann, müssen ein paar besondere Anforderungen an den Algorithmus gestellt werden. Zum einen muß es ein Stromchiffrierer sein, der einen kontinuierlichen Byte-Strom verarbeitet. Zum anderen muß er selbstsynchronisierend sein, da auf dem B-Kanal nur die Nutzdaten und keine zusätzlichen Steuersignale übertragen werden können. Außerdem kann es vorkommen, daß ein Byte ausfällt oder ein Byte zuviel erkannt wird.

Als Verschlüsselungsalgorithmus wird in dieser Arbeit der IDEA verwendet. Er läßt sich im Gegensatz zum DES, der für Hardware optimiert wurde, relativ einfach als Software realisieren. Mit seinem 128 Bit langen Schlüssel ist er sehr sicher. Es existieren $3,4 * 10^{38}$ mögliche Schlüsselkombinationen. DES mit 56 bittigem Schlüssel hat "nur" $7,2 * 10^{16}$ Kombinationen. Selbst wenn man es schafft, zehn Milliarden Schlüssel pro Sekunde auszutesten, dafür werden etwa 100000 schnelle PCs benötigt, dauert es etwa $5,4 * 10^{20}$ Jahre um den Schlüssel zu "knacken"! Beim DES wären es nur 41,7 Tage. Statt der PCs eignen sich FPGAs oder ASICs wesentlich besser zum Austesten von Verschlüsselungen. Gerüchteweise schaffen es gewisse Geheimdienste, einen DES-Schlüssel in 5 Minuten zu bestimmen, aber beim IDEA wäre es immer noch ein astronomischer Wert von $4,5 * 10^{16}$ Jahren. Der IDEA wird auch in vielen anerkannten Verschlüsselungsprogrammen wie PGP eingesetzt. IDEA ist ein Blockchiffrierer, der noch mit einem geeigneteten Rahmenalgorithmus zum Stromchiffrierer umfunktioniert werden muß. Eine gute Möglichkeit ist der "Cipher-Feedback Mode" (Abbildung 8).

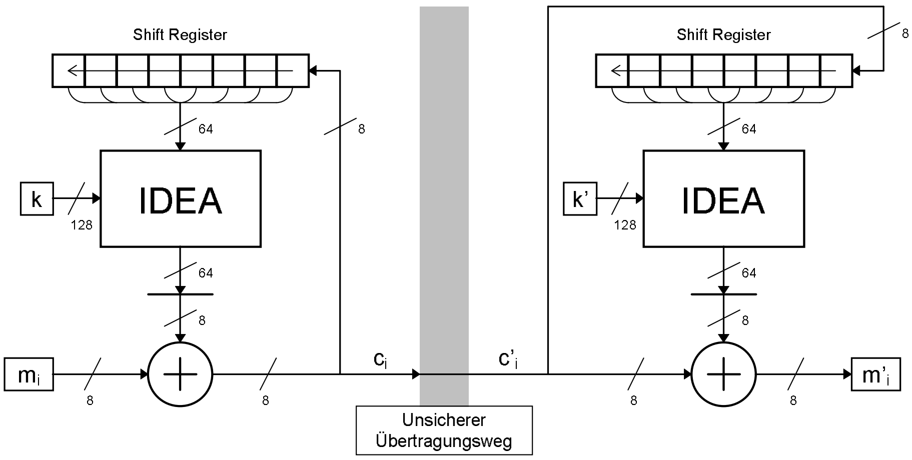


Abbildung 8: Cipher-Feedback Mode mit IDEA

Das wesentliche am Cipher-Feedback Mode ist, daß er die letzten acht verschlüsselten Bytes die übertragen werden, in einem Schieberegister sammelt. Damit hat er den 64Bit Eingangswert für die IDEA-Verschlüsselung. Da ja die verschlüsselten übertragenen Bytes auf beiden Seiten (Sender und Empfänger) in die Schieberegister gelangen, sind die Inhalte beider Schieberegister spätestens nach 8 Berechnugs-Zyklen ($8 * 125 * \mu s$ = 1ms) gleich und bleiben gleich, solange kein Übertragungsfehler auftritt. Somit ist der Ergebnis der IDEA-Verschlüsselung auf beiden Seiten gleich, wenn, voraussetzungsgemäß der Schlüssel (k) gleich ist. Von dem 64Bit IDEA-Verschlüsselungsergebnis wird ein Byte genommen, meist das MSB, und mit dem Byte aus dem Datenstrom verexclosivodert (Abbildung 8). Der einzige Nachteil dieses Cipher-Feedback-Algorithmus ist, daß die achtfache Menge an IDEA-Berechnungen durchgeführt werden muß, wie wenn der IDEA direkt eingesetzt würde.

Nun aber zum eigentlichen IDEA-Algorithmus (Abbildung 9). Der IDEA hat nur drei verschiedene Grundoperationen: XOR, Addition ohne Übertrag auf das 17. Bit und eine “IDEA-spezifische Multiplikation”. Zum XOR und der Addition gibt es nichts weiter zu sagen, aber zur IDEA-spezifischen Multiplikation. Diese Operation lautet “$x = (a * b) \bmod (2^{16} + 1)$”. Der Ausdruck $(2^{16} + 1)$ mit dem Resultat 65537 ist eine Primzahl. Folgende Festlegung ist zu beachten: Ist einer der Faktoren (a oder b) gleich Null, dann wird er zu 2^{16}, und wenn das Ergebnis x der IDEA-spezifischen Multiplikation gleich 2^{16} ist, wird es auf Null gesetzt.

Es existiert ein eleganter Weg, die Modulodivision durch 65537 zu umgehen und damit viel Rechenzeit und Speicher einzusparen. Nach der Multiplikation erfolgt eine Subtraktion des Low-Word vom High-Word des Produkts, und ist dabei ein Übertrag entstanden, wird eine Eins auf das Ergebnis addiert. Zur Veranschaulichung dient eine C-Funktion, die genau diese Operation durchführt (Abbildung 10).

Vor der Ausführung des IDEA-Algorithmus muß jedoch der Schlüssel expandiert werden. Dieser ist 128 Bit lang, aber der Algorithmus benötigt 52 mal den 16Bit Teilschlüssel (k1,1 - k1,6 / k2,1 - k2,6 / ... / k8,1 - k8,6 / k9,1 - k9,4). Der Ablauf gestaltet sich folgendermaßen:
Aus dem 128bitigem Schlüssel werden die ersten acht Teilschlüssel (k1,1 - k2,2) entnommen, dann erfolgt eine Rotation des Schlüssels um 25 Bit nach links, es werden wiederum die nächsten acht Teilschlüssel (k2,3 - k3,4) entnommen, und so weiter bis sämtliche 52 Teilschlüssel vorliegen. Zum Entschlüsseln werden die Teilschlüssel erstens in den acht Runden vertauscht und zweitens für die Teilschlüssel die inversen Elemente gesucht. Das bedeutet, für alle Teilschlüssel, die an der Addition beteiligt sind, ist es das 2er-Complement. Für die Teilschlüssel, die an der IDEA-Multiplikation beteiligt sind, wird das inverse Element mit dem “Erweitertem Euklidischen Algorithmus” ausfindig gemacht. In der vorliegenden Anwendung ist die IDEA-Entschlüsselung nicht von Bedeutung, weil beim CipherFeedback Mode auf beiden Seiten nur die Verschlüsselung erforderlich ist.

Noch eine Anmerkung zum IDEA. Dieser Algorithmus ist urheberrechtlich geschützt. Der kommerzielle Einsatz ist nicht erlaubt ohne Entrichtung entsprechender Lizenzgebühren. Aber für private und schulische Zwecke ist er frei verfügbar. Demnach darf das Endprodukt nicht mit diesem Algorithmus verkauft werden, aber erlaubt ist die (kostenlose) Verteilung der Software als Freeware. So ist es auch bei PGP, es ist Freeware.

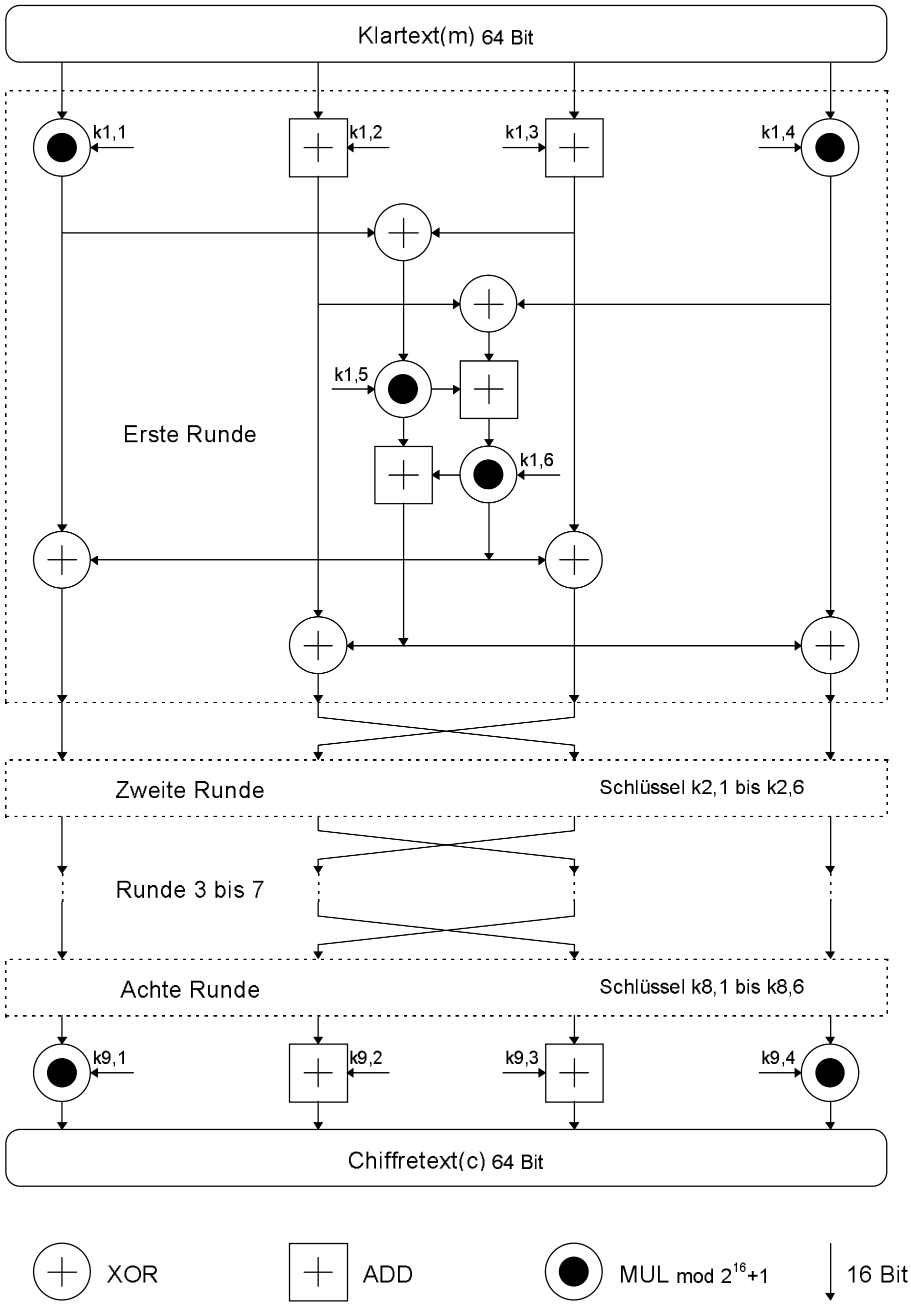


Abbildung 9: IDEA-Algorithmus

```
unsigned int idea_mul (unsigned int a, unsigned int b)
{
  unsigned long p;
  if( a==0 )
     return( 1-b );
  if( b==0 )
     return( 1-a );
  p=(unsigned long) a*b;
  b=(unsigned int) ( p );
  a=(unsigned int) (p>>16);
  return( ( b-a ) + ( b<a ) );
}
```

Abbildung 10: IDEA-spezifische Multiplikation in C

4.2 Blockschaltplan

In diesem Kapitel erfolgt die nähere Betrachtung der verwendeten Komponenten und deren Verschaltung im Verschlüsselungs-Telefon (Abbildung 11).

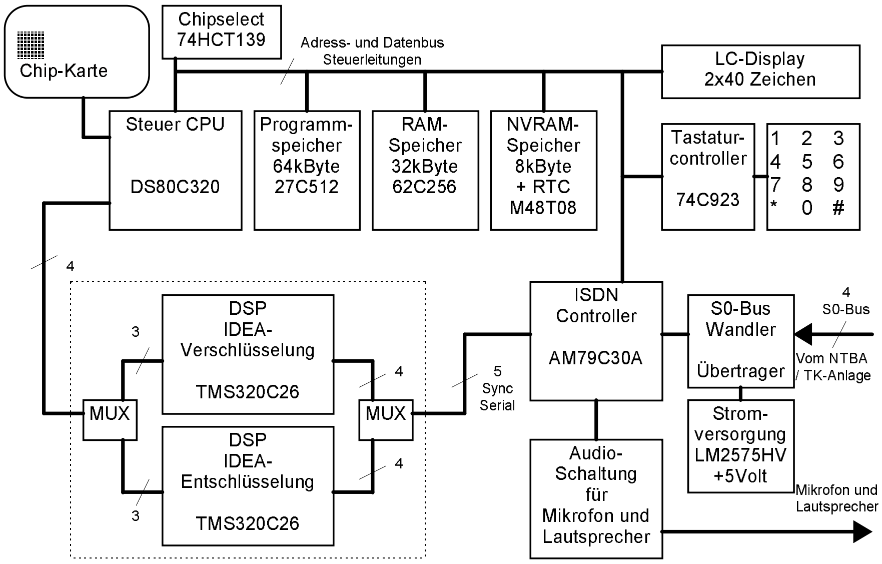


Abbildung 11: Blockschaltbild

Die Chipkarte enthält den Schlüssel für die Verschlüsselungsalgorithmen. Die Elektronik am Kartenleser wird so ausgelegt, daß sowohl eine Speicher- oder Prozessorkarten Verwendung finden kann. Die Karte wird mit einem Kartenleser am PC initialisiert und mit dem Schlüssel beschrieben. Wenn eine Prozessorkarte verwendet wird, kann sie durch eine PIN geschützt oder sogar Verschlüsselungsalgorithmen implementiert werden. Prozessorkarten sind aber wesentlich teurer als Speicherkarten, und sie sind kaum auf dem freien Markt verfügbar. Speicherkarten sind leichter zu beschaffen, wie z.B. die SLE4428, bieten aber kaum zusätzliche Funktionen und auch keine Sicherheit zum Schutz des Schlüssels gegen unbefugtes Auslesen.

Der Steuerprozssor in diesem Telefon ist ein 51er Derivat von Dallas, der einige deutliche Vorteile gegenüber dem normalen 80C51 hat. Der 80C320 verfügt über zwei serielle Ports. Der eine Port wird benötigt, um mit Prozessorchipkarten zu kommunizieren, und der andere zur Kontrolle der Ver- und Entschlüsselungs-DSPs. Die gesamte Software wird extern in einem 64kByte großem EPROM abgelegt. Dort liegen auch die Routinen für die DSPs, die der 80C320 beim Einschalten über einen seriellen Port zu den DSPs überträgt.

Da der Prozessor selbst nur über 256Byte RAM verfügt, wird noch 32kByte statisches RAM als externer Datenspeicher angeschlossen. Zusätzlich bekommt das Gerät noch 8kByte NVRAM. Das ist RAM, welches seinen Inhalt auch im spannungslosen Zustand behält, da es eine eigene Lithium-Batterie über dem Chip hat. In diesem Speicher können z.B. Telefonnummernlisten und Geräteeinstellungen gespeichert werden. Der Einsatz von NVRAMs mit eingebauter Echtzeituhr ist ebenfalls möglich.

Als Ausgabeeinheit zum Benutzer wird ein Flüssigkristall-Display eingesetzt, das direkt am Prozessor-Bus betrieben wird. Ein solches Display kann alle alphanummerischen Zeichen und sogar einige selbst definierte Zeichen darstellen. Zur Eingabe wird eine 16er oder 20er Tastatur eingesetzt, die über einen Tastaturcontroller mit dem Prozessor-Bus verbunden ist. Die Tastatur wird unter anderem über die zwölf bei einem Telefon üblichen Tasten (0 bis 9, * und #) verfügen. Zusätzlich kommen noch Tasten zur Menüsteuerung (Pfeiltasten, Löschen und Bestätigung) hinzu.

Das Wichtigste an einem ISDN-Telefon ist der ISDN-Controller-Baustein. Er hat die Aufgabe, den S0-Bus zu verwalten, die unteren Schichten des D-Kanals zu bedienen, die B-Kanäle zu verschalten und die A/D-D/A Wandlung der Sprachsignale zu übernehmen.

Der hier benutzte AM79C30A ist der einzige mir bekannte Baustein, der diese Anforderungen in einem Chip unterbringt. Er ist zudem relativ einfach zu beschaffen. Wichtig für die Verschlüsselung ist noch, daß B-Kanäle über eine extra synchronserielle Schnittstelle geleitet werden können. Diese wird benötigt, um die Daten der B-Kanäle zu den DSPs, die sie ver- und entschlüsseln, zu übertragen. Zum S0-Bus hin ist der Baustein über einen Übertrager verbunden, der die Signale galvanisch aus dem S0-Bus entkoppelt, da auch zusätzlich die Versorgungspannung von 40V mit auf dem S0-Bus liegt. Aus der 40V Versorgungsspannung auf dem S0-Bus wird mit Hilfe eines Schaltwandlers +5V erzeugt. Im gesamten Gerät wird nur diese eine Versorgungsspannug von +5V benötigt. Da es vorkommen kann, daß das Gerät mehr Leistung benötigt als der ISDN S0-Bus zur Verfügung stellen kann, besonders die DSPs haben einen hohen Stromverbrauch, muß es auch möglich sein, das Telefon mit einem externen Netzteil zu versorgen.

Ein Telefon kommt natürlich nicht ohne Mikrofon und Lautsprecher aus. Also muß noch eine Operationsverstärkerschaltung für die analoge Ein- und Ausgabe der akustischen Signale her. Der Audioteil des ISDN-Chips ist mit jeweils zwei Ein- und Ausgängen ausgestattet. Ein Ein- und Ausgangspaar wird für den Telefonhörer benutzt. Das andere Paar ist für Mikrofon und Lautsprecher im Telefongerät gedacht, der Freisprecheinrichtung. Zudem wird dem Lautsprecher im Gerät noch die Funktion der Klingel zugewiesen.

Die eigentliche Ver- und Entschlüsselung übernehmen die beiden DSPs von Texas Instruments. Sie werden mit 40MHz getaktet, das reicht aus, um je einen 8kByte/s großen Datenstrom mit dem IDEA-Algoritmus im Cipher-Block-Mode zu ver- oder entschlüsseln. Die DSPs haben keinen externen Datenspeicher und keinen direkten

Programmspeicher, sondern verfügen über ein $1,5k * 16Bit$ internes RAM. Das Programm für die DSPs wird nach dem Einschalten von der 51er-CPU über einen Bootloader in das RAM der DSPs geladen. Die DSPs befinden sich auf einer eigenen Platine, die auf die Hauptplatine aufgesteckt wird. Das geschieht aus zwei Gründen. Zum einen ist so der Algorithmus des Verschlüsselungsteils bei Bedarf leicht gegen einen anderen austauschbar. Und wären die DSPs mit auf die Hauptplatine gesetzt worden, hätte diese die Standard-Abmaße einer Eurokarte (160x100mm) weit überschritten.

5 Feinkonzept

Hier folgt nun eine detaillierte Beschreibung aller Hardwarekomponenten und eine genaue Beschreibung der Crypto-Software.

5.1 Die Chipkarte

Eine Chipkarte, ähnlich der Telefonkarte der Telekom, soll hier eingesetzt werden, um den geheimen Schlüssel zu speichern. In der ISO-7816 sind sämtliche Normen zum Thema Chipkarte enthalten [4]. Aber jeder Hersteller weicht für seine Anwendung von diesen Vorgaben ab.

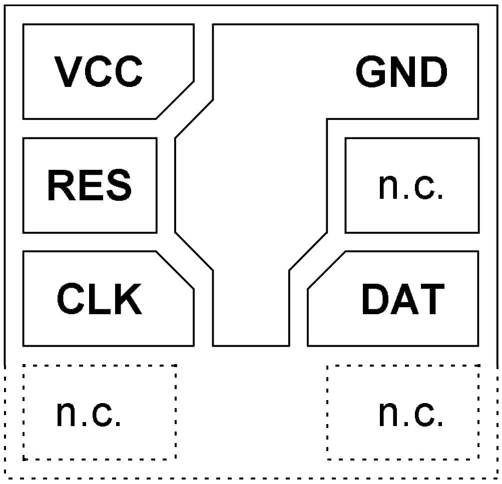


Abbildung 12: Kontaktbelegung einer üblichen Chipkarte

Die Belegung der Kontakte auf Chipkarten (Abbildung 12) ist fast immer gleich, sogar die von Speicher- und Prozessorkarten. Die Kontakte für die Stromversorgung (VCC und GND) befinden sich immer an derselben Stelle. Sämtliche Karten arbeiten mit +5V, die hier über den Schutzwiderstand R60 (siehe Stromlaufplan ab Seite 63) geführt werden. Ein in der Entwicklungsphase vorgesehener Kontakt für die EEPROM-Programmierspannung kommt nicht zum Einsatz, da alle Chipkarten mit einer eigenen Ladungspumpe ausgestattet sind. Der bedeutendste Unterschied in der Beschaltung einer Speicher- oder Prozessorkarte betrifft den Takteingang (CLK). Bezüglich der Speicherkarte ist er der Schiebetakt für den seriellen Bitstrom am Datenkontakt (DAT). Mit jeder fallenden Flanke am Takteingang wird das nächste Bit am Datenausgang ausgegeben. Für die Prozessorkarte ist der CLK-Kontakt der Eingang für den Prozessortakt, der meist 3,57MHz beträgt. Er wird benötigt, weil eine Prozessorkarte über keine eigene Takterzeugung verfügt; denn eine Chipkarte ist zu dünn für einen Quarz. Also muß die Hardware des Chipkartenschachts so ausgelegt sein, daß sie eine Frequenz von 3,57MHz am CLK-Kontakt zur Verfügung stellt oder den Pegel des CLK-Kontaktes direkt festlegen kann. Dafür ist IC10a/b/c zuständig; es erzeugt die Frequenz mit Hilfe eines Quarzes, und der Prozessor kann über zwei Portleitung den CLK-Pin high, low oder mit dieser Frequenz beschalten.

Mittels des Reset-Kontaktes (RES) auf der Prozessorkarte wird er ein low-aktiver

P1.5	P1.7	CC-Clk
0	0	1
0	1	0
1	0	1
1	1	3,57 MHz

Abbildung 13: Clk-Pin der Chipkarte

Hardwarereset ausgelöst. Auf der Speicherkarte hat dieser Eingang noch zusätzliche Aufgaben, die in Verbindung mit dem CLK-Eingang bestimmt werden. Aber für die Hardware des Chipkartenschachts, der an die Steckleiste X3 angeschlossen ist, reicht es, ihn mit einer Portleitung des Prozessors zu verbinden. Der gesamte bidirektionale Datenaustauch findet über den Open-Collector DAT-Kontakt statt, ähnlich dem RS232 Protokoll, jedoch mit 5Volt Pegel und nur einer Leitung für beide Richtungen. Es existieren keine zusätzlichen Steuerleitungen; die Übertragung enthält ein Startbit(0), 8 Datenbits, ggf. ein Paritätsbit und 2 Stopbits(1). Die Baudrate ergibt sich durch eine Teilung des Prozessortaktes am CLK-Eingang, die meist 372 beträgt. Damit ergeben sich bei 3,57MHz 9600 Baud. Deswegen wird ein serieller Port des Prozessors für die Kommunikation mit der Chipkarte verwendet. Da der Prozessor jeweils eine extra Leitung für beide Richtungen hat, werden sie einfach verbunden. Es existiert eine Unmenge genormter und ungenormter Protokolle für Prozessorchipkarten, so daß sie nicht weiter erwähnt werden, sondern nur das hier eingesetzte Protokoll, welches eine reduzierte Variante des ISO7816-T=1 ist (Abbildung 14). Nach dem Reset sendet die Karte einen ATR-String, der Inhalt dieser Bytefolge gibt Auskunft über das verwendete Protokoll und ist in der ISO7816 Norm definiert. Für die Chipkarte handelt es sich nur um einen konstanten String, der gesendet wird.

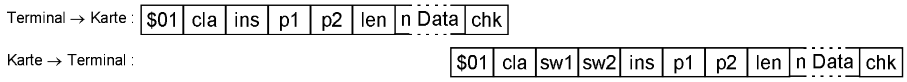


Abbildung 14: Protokoll der Prozessorchipkarte

Danach wartet die Karte auf Daten vom Terminal. Ein Paket vom Terminal beginnt immer mit einem 6 Byte langen Header, der ausschließlich mit $01 beginnt. Dieses Byte war vorgesehen, um mehrere Karten am Bus zu adressieren, aber hier gibt es nur eine einzige. Das zweite Byte ist das Class-Byte und wählt die Anwendung innerhalb der Karte aus. Da hier nur eine Anwendung existiert, nämlich die Bereitstellung des Schlüssels, gibt es auch nur ein Classbyte, welches willkürlich den Wert $02 erhält. Dann folgt das Instruction-Byte mit dem Kommando an die Karte. Die Commandbytes sind ebenfalls in der ISO7816-4/-5 genormt, aber kaum jemand hält sich an die Regelung. P1 und P2 sind zusätzliche Parameter für das Kommando. Als letztes Byte im Header erscheint die Länge der Daten, die das Terminal mit diesem Kommando an die Karte schicken will. Mit dem Inhalt $00 werden keine Daten zur Karte geschickt, und das Datenfeld ist leer. Zum Abschluß des Packets kommt ein Prüfbyte, das als Checksumme alle Bytes des Pakets verexclusivodert enthält. Hat die Karte nun ein solches Packet als gültig erkannt, so schickt es ein Antwortpaket, welches gleichfalls aus einem Header, ggf. Daten und einem Prüfbyte besteht. Das Class-Byte, Instruction-Byte und die beiden Parameterbytes sind identisch mit denen des Headers vom Terminal. Zusätzlich enthält aber dieser Header noch zwei Statuswortbytes, die Fehlercodes oder ähnliches enthalten, vergleichbar dem Return-Wert bei C-Funktionen. Jetzt macht das Len-Byte eine Aussage über die Anzahl der Datenbytes, die die Karte ans Terminal zu schicken beabsichtigt. Abgeschlossen wird das Paket wieder mit dem Prüfbyte.

Die einfache Version der hier verwendeten Karte wird nur drei Kommandos kennen: Status anzeigen ($F2), PIN vergleichen ($20) und Schlüssel ausgeben ($C0). Da es schwer ist, als Privatkunde auf dem freien Markt Prozessorchipkarten zu kaufen, fertigt man sich einfach eine 0,7mm dünne Leiterplatte in Form einer Chipkarte (85 x 54mm) und lötet einen SMD-Einchip-Mircocontroller auf. Hier wird ein PIC16C84 von der Firma Microchip verwendet, da er über EEPROM Speicher verfügt. Möglicherweise wird jedoch die Verfügbarkeit von Prozessorchipkarten in Zukunft etwas einfacher. Es folgt nun ein Beispiel für einen Protokollablauf (Abbildung 15).

```
ATR  : 3B 80 00                ; Answer to Reset

T>C  : 01 02 F2 00 00 00       ; Terminal fordert Status an
       F1
C>T  : 01 02 90 00 A4 00 00 03 ; Karte gibt 3 Byte Status
       01 00 03
       36

T>C  : 01 02 20 00 01 02       ; Terminal gibt Karte PIN Nummer
       12 34
       06
C>T  : 01 02 90 00 20 00 01 00 ; Karte bestaetigt PIN
       B2

T>C  : 01 02 C0 00 01 00       ; Terminal fordert Schluessel an
       C2
C>T  : 01 02 90 00 C0 00 01 10 ; Karte gibt 16 Byte Schluesssel aus
       .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
       ..
```

Abbildung 15: Kartenprotokoll Beispiel

Im Rahmen der vorliegenden Diplomarbeit, hinsichtlich der zur Verfügung stehenden Zeit, ist die Anzahl der Funktionen begrenzt. Ein erweiterter Funktionsumfang würde zudem folgende Optionen ermöglichen: PIN ändern, mehrere Schlüssel verwalten, RSA-Algorithmen.

5.2 Der Steuerprozessor mit Speicher

Der 51er-Prozessor hat einen 8Bit Datenbus und 16Bit Adreßbus. Aus fertigungstechnischen Gründen (nur 40 Pins im Gehäuse!) werden die acht Datenbits mit den unteren acht Bits des Adreßbusses gemultiplext [3]. IC2 (74ACT573) speichert bei fallender Flanke von ALE das Low-Byte der Adresse zwischen. Der 51er-Prozessor kennt drei Arten von Speicher. Das interne RAM ist 256 Bytes groß und enthält zudem 128 Special-Function-Register (SFRs). In diesem Bereich wird zudem der Stack gehalten. Der Programmspeicher des 51er Prozessors kann nur gelesen werden. In ihm stehen nur die Opcodes mit ihren Parametern und konstante Daten. Mit Hilfe des /EA-Pins wird festgelegt, ob interner oder externer Programmspeicher angesprochen werden soll. Da aber der verwendete DS80C320 (IC1) über keinen internen Programm-Speicher verfügt, wird Pin /EA fest auf Masse gelegt. Die /PSEN-Leitung des Prozessors steuert das Auslesen des externen Programmspeichers, einem EPROM IC3 (27C512), welches zuvor mit einem EPROM-Brenner mit den Daten beschrieben wird. In diesem Speicher befindet sich auch das Programm für die beiden DSPs und wird nach jedem Einschalten an sie gesendet. Der externe Datenspeicherbereich (Abbildung 16) kann vom Prozessor gelesen und beschrieben werden, ist wie der Programmspeicherbereich 64kByte groß und wird mit den /RD und /WR Leitungen angesprochen. Mittels IC8a (74HCT139) und IC9a (74HCT00) wird der Datenspeicherbereich in eine 32kByte und vier 8kByte Sektionen geteilt. Der 32kByte-Bereich ist bestimmt für IC3 (62C256), einem statischen RAM. Das Chipselect für dieses RAM ist ausschließlich die Adreßleitung A15; sie ist für die Adressen \$0000 bis \$7FFF low, und der eigentliche Lese- oder Schreibzugriff erfolgt dann mit den Steuerleitungen /RD oder /WR. Wenn A15 high ist, wird IC8a (74HCT139) freigegeben, das den Adreßbreich von \$8000 bis \$FFFF vierteilt. Die ersten 8kByte ab \$8000 sind für einen besonderen RAM-Baustein (IC5) vorgesehen. Er ist durch eine Batterie gebuffert und enthält in den letzten 8Byte eine Echtzeituhr (RTC). Diese Echtzeiteinrichtung ist nicht nur als Uhr für den Benutzer gedacht, sondern es können zudem Timestamps für die Verschlüsselung erzeugt werden (z.B. jeden Tag einen anderen abgeleiteten Schlüssel). Die übrigen drei 8kByte Bereiche im Datensektor sind für die Peripherie vorgesehen, wie LC-Display, Tastatur und ISDN-Controller.

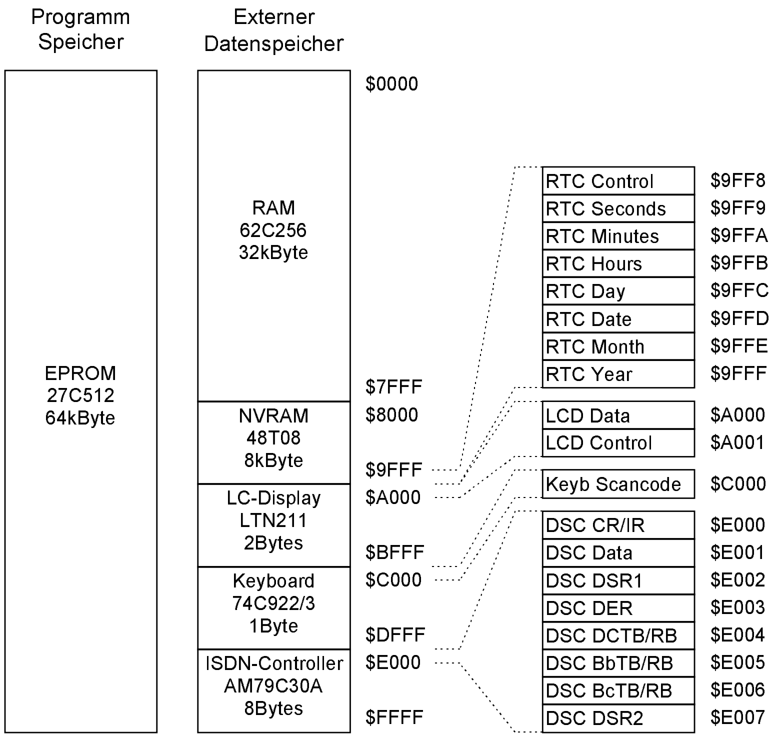


Abbildung 16: Speicherraum des Steuerprozessors

5.3 Der Steuerprozessor mit Peripherie

Am Systembus des Steuerprozessors befinden sich die drei zuvor erwähnten Peripherieeinheiten. LCD-Module gibt es in vielen verschiedenen Ausführungen: mit oder ohne Hintergrundbeleuchtung, unterschiedliche Spalten- und Zeilenzahlen, verschiedene Zeichengrößen. Aber die Hardwareanbindung und Softwareschnittstellen sind fast immer gleich, bedingt durch einen Quasi-Standard seitens der unterschiedlichen Hersteller, die aus dem Angebot der verfügbaren Controller wählen müssen. An der 14poligen Steckleiste (X1) befinden sich acht Datenleitungen, eine Adreßleitung, eine Schreib/Leseleitung, ein Enable und drei Leitungen für Versorgungsspannung und Kontrast. Alle Steuer-, Adreß- und Datenleitungen können direkt mit dem Systembus verbunden werden, mit Außnahme der Enableleitung. Diese darf nur in den High-Zustand gehen, wenn /RD oder /WR low sind und die Adresse $A000-$BFFF anliegt. Es werden vom Display zwar nur zwei Bytes benötigt, aber es sind für jeden Peripheriebaustein 8kByte vorgesehen. Der Aufwand wäre im Sinne der Aufgabenstellung, einfach und preisgünstig, zu hoch, um die Adressen für alle Peripheriebausteine aufs Byte genau auszudecodieren, denn dann wäre eine Vielzahl von weiteren TTL-Bausteinen oder ein programmierter Logikbaustein erforderlich. Der eingebaute Controller des LC-Displays hat zwei Register. In das erste Register, das Datenregister, wird der ASCII-Code des darzustellenden Zeichens geschrieben. Das zweite Register, das Controllregister, dient zur Steuerung der Displayfunktionen, wie Anzeige löschen, Cursorposition setzen, eigenen Zeichensatz definieren u.s.w.

Als weiteres Peripheriegerät für den Anschluß an den Systembus ist eine Tastatur mit 16 oder 20 Tasten vorgesehen. Der Tastaturcontroller ist auf der Hauptplatine untergebracht und wird über den Steckverbinder X2 mit der Tastatur-Tastenmatrix verbunden. Der hier eingesetzte Tastaturcontroller 74C922/3 von National Semiconductors übernimmt alle für den Betrieb einer kleinen Tastaturmatrix nötigen Aufgaben. Dazu gehören die Beschaltung der Spaltenleitungen und die Auswertung der Zeilenleitungen. Darüberhinaus übernimmt er auch die Entprellung der Tasten. Ist eine Taste betätigt worden, löst er einen Interrupt aus, und der Prozessor kann sich den Scancode der zuletzt gedrückten Taste aus dem Register des Tastaturcontrollers holen.

5.4 Die ISDN-Hardware

Das letzte und wichtigste Peripheriegerät betrifft die ISDN-Hardware (Abbildung 17). Alle für den ISDN-Bus und das Audio-Interface nötigen Funktionen übernimmt das IC6 (Am79C30A), welches sich der CPU mit nur 8 Registern zeigt. Es existieren insgesamt über sechzig Register in diesem Chip, die jedoch indirekt angesprochen werden. Die direkt adressierbaren Register sind fast alle für die FIFO-Buffer der B-Kanäle und des D-Kanals vorgesehen. Es ist ein Command- und Datenregister für die indirekt adressierten Register vorhanden. Das Interrupt- und D-Kanal-Statusregister sind ebenfalls direkt adressierbar. Alle weiteren Register des Chips werden über das Command- und Datenregister angesprochen. Durch das Schreiben der Registernummer in das Commandregister wird mit dem Datenregisterzugriff der Zugang zum entsprechenden Registerinhalt ermöglicht.

Der S0-Bus wird über ein Übertragerpaar (TR1) an den ISDN-Chip angeschlossen. Die Dioden (D31-D38) schützen ausschließlich vor Überspannung. Da der S0-Bus zusätzlich eine Spannung von 40V führt, wird diese für den Betrieb des Geräts verwendet. Die 40V werden an den Mittelanzapfungen des ISDN-Übertragerpaars auf der S0-Busseite abgegriffen. Am Jumperfeld JP1 kann festgelegt werden, ob und mit wel-

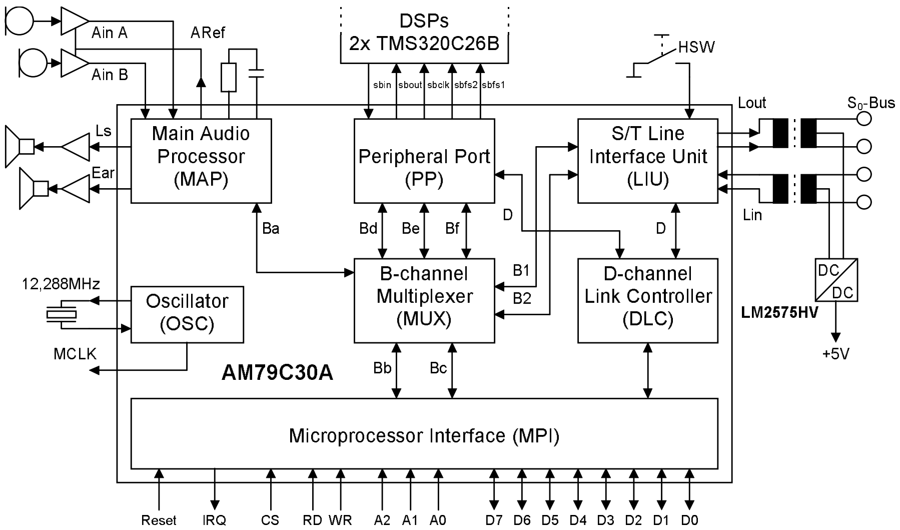


Abbildung 17: ISDN-Hardware

cher Polarität die Spannung genutzt wird. Die Polarität spielt eine Rolle hinsichtlich der Notstromberechtigung des Gerätes. Der eingesetzte Step-Down-Wander (IC13) kann Spannungen im Bereich von 8V bis 57V in 5V (1A max.) umsetzen. Über die Diode D10 kann der Spannungsanschluß vom S0-Bus erfolgen, und an Diode D11 kann wahlweise auch ein externes Netzteil angeschlossen werden. Besondere Aufmerksamkeit wird der Induktivität L10 gewidmet. Diese Spule muß erstens ihre Induktivität (1mH) auch bei 52kHz halten, und zweitens ist die Auslegung des Spulendrahtes für mindestens 2A unumgänglich. Da der ISDN-Chip die Audiofunktionen übernimmt, hat er analoge Ein- und Ausgänge für je zwei Mikrofone und Lautsprecher. Also gibt es auch je zwei Mikrofon- und Lautsprecherverstärker.

Die Analogeingänge des ISDN-Chips erwarten eine Eingangsspannung von $\sim 0,625$ V_{rms} bei 0dB. Die Mikrofonverstärker (IC11a/b) sind OPVs in invertierendem Betrieb und liefern eine ein- bis fünfzig Verstärkung, die mit dem Poti P3/P4 einstellbar ist. Ebenso können auch andere Audioquellen (z.B. Kassettenrecorder) an das Telefon angeschlossen werden. Die Mikrofone werden mit dem Steckverbinder X6 verbunden. Zum Audioeingang gehört auch die RC-Kombination R30 und C30, die für den A/D-Wandler im ISDN-Chip gebraucht wird. Der ISDN-Chip liefert auch eine Referenzspannung an Pin 43, die ca. +2,5V beträgt und als Null-Spannung für die OPVs genutzt wird.

Die Audioausgänge des ISDN-Chips sind als Differenzialausgänge ausgelegt und liefern $\sim 1,25V_{rms}$ bei 0dB. Diese Ausgänge sollten auch im Differenzsignalbetrieb genutzt werden, um Störungen zur unterdrücken. Daher werden die OPVs (IC11c/d) auch als Differenzverstärker eingesetzt. Weiterhin existiert noch eine Lautsprecherendstufe (IC12), die niederohmige Lasten $(4\Omega/8\Omega)$ vertragen kann und eine Leistung von einem Watt für die Lautsprecher liefert. Die Lautstärke läßt sich am Poti P1/P2 einstellen. Angeschlossen werden die Lautsprecher am Steckverbinder X5.

Der Steckverbinder X7 steht zur Verfügung für den Anschluß eines Tasters oder Schalters, der dem Telefon anzeigt, ob der Hörer grade aufgelegt oder abgehoben ist. Eine Flanke auf dieser Leitung kann einen Interrupt auslösen.

Für die Verschlüsselung der Daten ist der serielle Peripherie-Port des ISDN-Chips wichtig, denn er kann drei B-Kanäle synchron seriell übertragen, die mit dem Multiplexer im ISDN-Chip auswählbar sind. In der vorliegenden Arbeit werden jedoch nur zwei B-Kanäle benötigt, einer zum S0-Bus hin und ein weiterer in Richtung A/D-D/A-Wandler. Dieser serielle Port des ISDN-Chips wird über den Steckverbinder X12 mit der DSP-Platine verbunden.

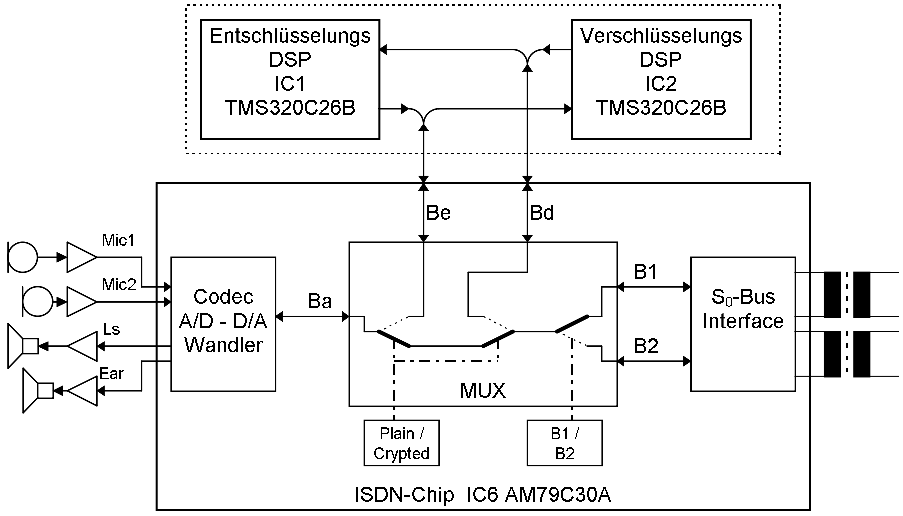


Abbildung 18: Anbindung der DSPs an den ISDN-Chip

Auf die Steckleisten X11 und X12 wird die DSP Platine aufgesetzt. Über X11 sind die DSPs mit der CPU (IC1) verbunden. Diese Verbindung dient zum Laden des Programms und zur Kontrolle der DSPs durch die CPU. Über X12 sind die seriellen Schnittstellen der DSPs direkt mit dem ISDN-Chip verbunden; ein Datenfluß des B-Kanals vom ISDN-Chip zu den DSPs und zurück ist möglich (Abbildung 18). Jedoch nur durch die Unterstützung des Multiplexers im ISDN-Chip: Zuerst wird einer der beiden B-Kanäle vom S0-Bus-Interface ausgewählt, eben der B-Kanal, der einem von der Vermittlungsstelle über den D-Kanal zugeteilt wurde. Falls das Gespräch nicht verschlüsselt ist, wird dieser Kanal direkt zum Audioteil durchgeschaltet. Aber im Fall eines verschlüsselten Gespräches wird nun der B-Kanal vom S0-Bus-Interface zum Bd-Kanal der seriellen Peripherieschnittstelle und der Be-Kanal der seriellen Peripherieschnittstelle zum Audioteil durchgeschaltet. Auf dieser seriellen Peripherieschnittstelle werden hintereinander drei B-Kanäle übermittelt. Der dritte B-Kanal Bf wird hier nicht benutzt. Diese Schnittstelle (Abbildung 19) hat fünf Leitungen, die am Steckverbinder X12 anliegen.

Es gibt zwei Frame-Impulse, eine Clockleitung und je eine Datenleitung für jede Richtung. Bis auf SB-In kommen alle anderen Signale vom ISDN-Chip. Normalerweise wechseln die Daten bei der steigenden Flanke der Clockleitung, eine Umprogrammierung auf die fallende Flanke im ISDN-Chip ist jedoch vorgesehen. Diese Möglichkeit wird in der vorliegenden Arbeit genutzt, um eine Kompatibilität zur Schnittstelle der DSPs zu schaffen. Die DSPs verfügen über je zwei Frame-Impulseingänge, einer zum Empfangen und einer zum Senden. Der besondere Trick an der Verschaltung der DSPs mit ISDN-Chip ist nun, daß die beiden Frame-Impulsleitungen des ISDN-

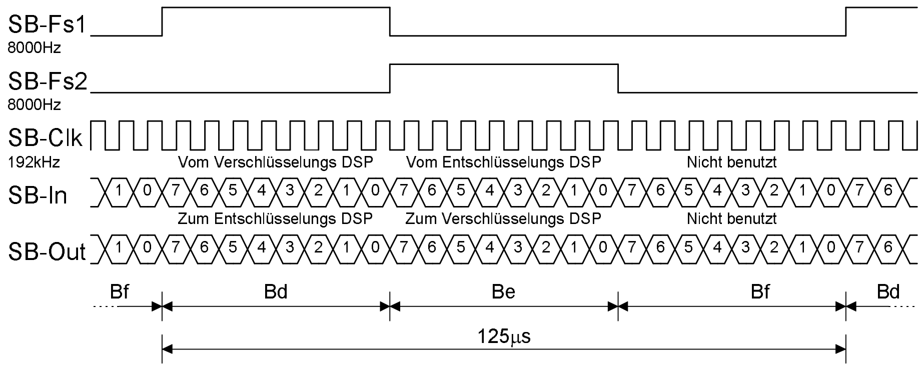


Abbildung 19: Signale zwischen ISDN-IC und DSPs

Chips direkt an die beiden Frame-Eingänge des DSPs gehen, aber beim zweiten DSP vertauscht sind. Damit empfängt der Verschlüsselungs-DSP Daten vom Be-Kanal und sendet auf dem Bd-Kanal. Und der Entschlüsselungs-DSP empfängt vom Bd-Kanal und sendet auf dem Be-Kanal. Für die DSPs werden nur noch die Framesignale mit Hilfe eines 74HCT86 invertiert.

5.5 Die Ver- und Entschlüsselungs-DSPs

Die Software für die DSPs hat ausschließlich die Aufgabe, das eintreffende Byte mit dem IDEA-Algorithmus im Cipher-Feedback-Mode zu ver- oder entschlüsseln und es wieder auszugeben. Dabei unterscheiden sich die Programme für das Verschlüsseln und das Entschlüsseln nur geringfügig voneinander. Der einzige Unterschied betrifft die Quelle des ankommenden Bytes, welches in das Schieberegister geschoben wird. Da die beiden Schieberegister auf beiden Seiten mit denselben Daten gefüllt werden sollen, wird immer das übertragene verschlüsselte Byte genommen. Beim Verschlüsseln ist es das Byte nach der Verexclusivoderung bzw. beim Entschlüsseln das Byte vor der Verexclusivoderung. Ansonsten sind beide Programme identisch (Abbildung 20).

Das vollständige Programm wird in einer Interrupt-Service-Routine laufen, ohne jegliche Beteiligung des Hauptprogrammes.

Da der IDEA-Algorithmus 8,5 Runden (die halbe Runde bezieht sich auf die Operation mit dem Teilschlüssel k9.x) hat, wird er nicht einfach linear geschrieben. Es bietet sich an, eine Zählschleife von 8 Durchgängen zu programmieren und die letzte halbe Abschlußrunde isoliert zu schreiben. Der Zeitverlust durch die Programmierung der Schleife ist unkritisch und in Hinsicht auf den eingesparten Programmspeicher lohnenswert.

Bezüglich des Programmcodes ist von den drei Grundoperationen des IDEA wieder nur die IDEA-spezifische Multiplikation interessant. Eine einfache Addition und XOR-Opreration braucht nicht weiter erläutert zu werden, da sie nur jeweils aus einem Assemblerbefehl bestehen. Die IDEA-spezifische Multiplikation (Abbildung 21) wird analog zum C-Code (Abbildung 10) programmiert. Das vollständige DSP-Listung ist ab Seite 57 abgedruckt.

Zum IDEA gehört auch das Zerlegen des Schlüssels in die 52 Teilschlüssel. Diese Aufgabe kann sowohl der DSP als auch die CPU übernehmen, da sie nur einmal am

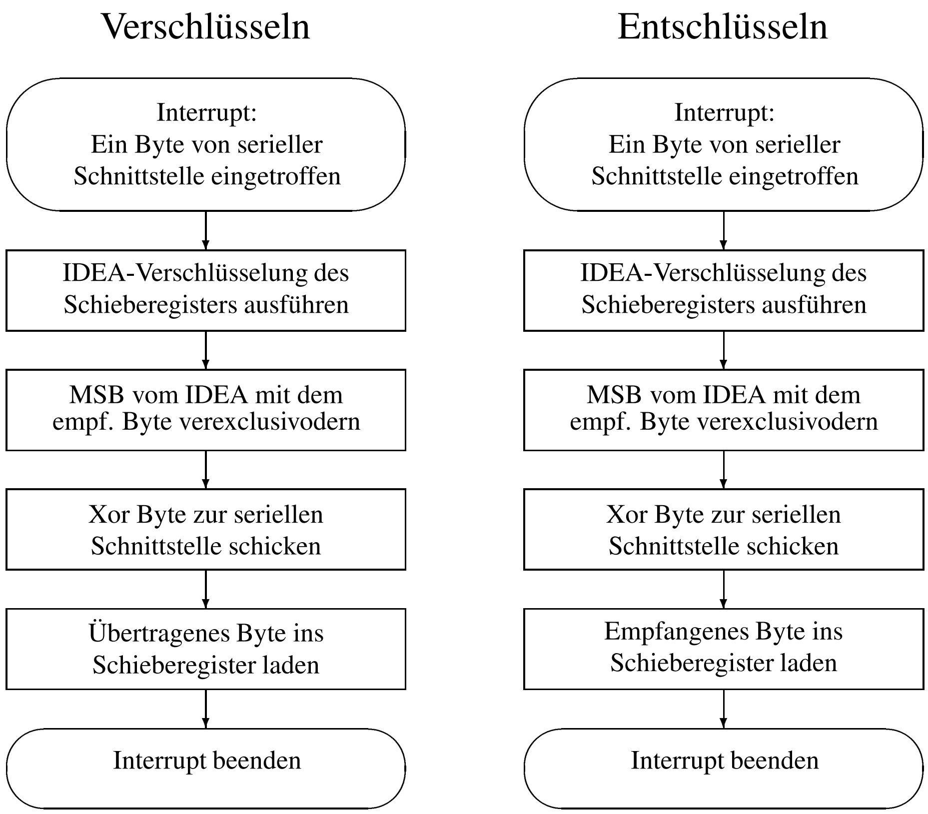


Abbildung 20: Die Interrupt-Routinen der DSPs

Anfang bei der Initialisierung des Schlüssels durchgeführt wird. In der vorliegenden Arbeit wird der DSP diese Aufgabe mitübernehmen, weil der DSP effektiver mit 16Bit-Werten umgehen kann. Desweiteren ist in "kryptographischer Hinsicht" darauf zu achten, daß das Schieberegister für den Cipher-Feedback-Mode im Verschlüsselungs-DSP immer mit anderen beliebigen Werten initialisiert wird. Wenn es immer mit denselben Werten (z.B. $00) initialisiert würde, gäbe es für einen Angreifer unter Umständen Angriffsmöglichkeiten. Die Initialisierungsaufgabe kann auch gleich von der CPU beim Programmladen in die DSPs mitübernommen werden. Es können die Zufallszahlen des ISDN-Chips genutzt werden oder beispielsweise ein angelegter Zähler im batteriegepuffertem RAM zum Einsatz kommen.

Zur Programmladung der DSPs soll folgendes erwähnt werden. Da die DSPs über kein eigenes (E)PROM verfügen, muß das Programm nach dem Einschalten der Versorgungsspannung in das interne RAM der DSPs geladen werden. Dafür werden vom DSP drei Leitungen benötigt: /Reset, XF und BIO. Der /Reset ist der lowaktive Hardwarereset des DSP, und nach der steigenden Flanke ist der DSP bereit, Daten entgegenzunehmen. XF ist der Statusausgang des DSP und zeigt an, ob die Daten korrekt empfangen worden sind. Für den seriellen Dateneingang ist BIO zuständig. Ähnlich der RS232-Schnittstelle gibt es hier auch hier, wie schon bei der Chipkarte, ein Startbit (0), acht Datenbits und ein oder mehr Stopbits (1). Die Baudrate wird vom DSP anhand des "Baud Detect Word" automatisch erkannt. Nach dem Resetvorgang des DSP schickt der Steuerprozessor eine bestimmte Bytefolge, wie Abbildung 22 sie zeigt.

Das "Baud Detect Word" dient dem DSP zur Ausmessung der Baudrate. Gesendet wird hier einfach $FF. Das "Status Word" enthält nur den High-Teil der Trans-

```
idea_mul:
          LAC  *             ; Lade Teilschluessel
          BZ   idea_10       ; Ist Er = 0 ? , dann spring zu idea_10
          LAC  data          ; Lade Eingangsdaten
          BZ   idea_11       ; Ist Er = 0 ? , dann spring zu idea_11
          LT   data          ; Lade Eingangsdaten ins TM-Register
          MPYU *+            ; Multipliziere Teilschluessel mit TM,
                             ; und erhoehe den Schluesselzeiger
          PAC                ; Lade Product in Accu
          SPH  temp          ; Speichere High-Word des Produkts in temp,
          SUBH temp          ; Subtrahiere High-Teil des Accus mit temp,
                             ; also High-Teil des Accus loeschen
          SUB  temp          ; Subtrahiere temp vom Accu,
                             ; also High-Teil vom Low-teil des Products
          BNC  idea_12       ; Ist Carry = 0 ?, dann spring zu idea_12
          B    idea_13       ; Springe zu idea_13
idea_10:  LAC  data          ; Lade Eingangsdaten
          B    idea_14, *+   ; Spring zu idea_14
                             ; und erhoehe den Schluesselzeiger
idea_11:  LAC  *+            ; Lade Teilschluessel
                             ; und erhoehe den Schluesselzeiger
idea_14:  NEG                ; Negiere Accu
idea_12:  ADDK 001h          ; Addiere 1 auf Accu
idea_13:  SACL data          ; Schreibe Daten zurueck
```

Abbildung 21: IDEA-Multiplikation in DSP-Assembler

ferlänge und einige Konstanten. Gesendet wird in der vorliegenden Arbeit "00001xxx". Das "Interrupt Word" enthält in den obersten zwei Bits die Speicherkonfiguration des DSP und in den unteren sechs Bits die Interruptfreigaben. In dieser Anwendung wird \$40 geschickt, da die Interruptfreigabe dann durch das Programm selbst erfolgt. Das "Length" Byte enthält den Low-Teil der Transferlänge. Dann folgt der eigentliche Datentransfer. Da der DSP ein 16Bit Prozessor ist, müssen immer zwei Bytes pro Speicherplatz übertragen werden. Es wird die Anzahl der 16Bit Worte übertragen, die in der Transferlänge angegeben wurde. Abschließend werden noch zwei Byte Checksumme übertragen, die 16Bit Summe aller Programmdaten. In der Checksumme gehen die Kontrolwörter nicht mit ein. Ist nach dem Übertragen der Checksumme der XF-Pin des DSPs immer noch high, ist alles in Ordnung. Geht er auf low, ist dieses ein Indiz für einen Fehler in der Übertragung. In dem Fall sollte der DSP in den Resetzustand gebracht und der Programmtransfer erneut probiert werden. Bei der nächsten fallenden Flanke am BIO-Pin, erzeugt durch das Senden von \$FF, wird das transferierte Progamm gestartet. Die Leitungen XF und BIO unterliegen dann der Programmkontrolle und sind somit für den Programmierer frei verfügbar. In dieser Arbeit wird das nicht genutzt.

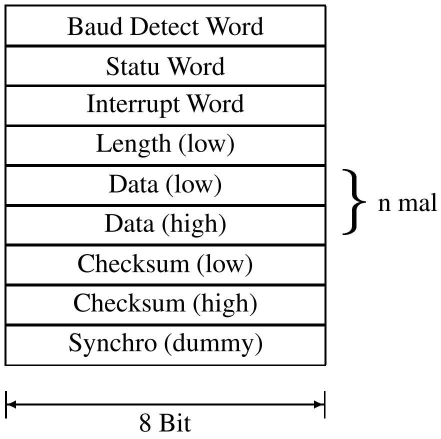


Abbildung 22: Bootloader Protokoll für die DSPs

6 Zusammenfassung

Abschließend erfolgt eine kurze Zusammenfassung der Arbeit mit einer eigenen Bewertung. Desweiteren werden einige Punkte aufgezeigt, wo und wie dieses Gerät verbessert und erweitert werden kann.

6.1 Bewertung

Die in der Aufgabenstellung genannten Punkte werden in der vorgelegten Arbeit erfüllt. Der Datenstrom läßt sich in der vorgesehenen Zeit von $125\mu s$ verschlüsseln.

Bei einem Verlust der Synchronisation wird nach 1ms die Synchronisation wieder hergestellt. Durch die Wahl eines 128bit Schlüssels ist eine zufällige Erkennung weitgehend ausgeschlossen.

Zum größten Problem in dieser Diplomarbeit wurde im Grunde genommen genau das, was ursprünglich nicht Teil der Arbeit war: der ISDN D-Kanal. Seine Implementation war für eine getrennte, parallel verlaufende Diplomarbeit vorgesehen. Aber aus gesundheitlichen Gründen war der Kommilitone gezwungen, seine begonnene Diplomarbeit abzubrechen, und damit gab es für das Verschlßselungstelefon auch kein D-Kanal-Protokoll. Und ein ISDN-Telefon ohne D-Kanal-Protokoll kann keine Verbindung zur Vermittlungsstelle und damit auch nicht zu einem anderen Telefon aufbauen. Wenn also der Verschlüsselungspartner fehlt, wird auch die Verschlüsselung als solche überflüssig. Daher mußte noch das ISDN D-Kanal Protokoll notdürftig in den Steuerprozessor implementiert werden.

Diese zusätzliche Arbeit hat viel von den zeitlichen Ressourcen der Diplombearbeitungszeit verbraucht, so daß nicht mehr die letzten noch erforderlichen Arbeiten an diesem Gerät durchgeführt werden konnten. Das hier programmierte D-Kanal Protokoll arbeitet nur provisorisch und sollte nicht ohne weitere Verbesserungen eingesetzt werden.

Probleme gab es auch mit dem Rauschen der analogen OP-Verstärkerschaltung, das vom Digitalteil der Schaltung verursacht wurde. Das Rauschverhalten müßte noch verbessert werden, aber die Bearbeitungszeit reichte nicht dafür aus.

Alles rund um die Kryptographie und den DSPs funktioniert hinreichend gut und macht keinerlei Probleme. Auch die gesamte digitale Hardware des ISDN-Telefons lief auf Anhieb.

6.2 Ausblicke

Man kann dieses Verschlüsselungstelefon in vielerlei Hinsicht verbessern und ausbauen. Aber zu den wichtigsten Erweiterungen gehört ein Public-Key-Algorithmus, wie RSA und Diffie-Hellmen-Schlüsselaustausch. Beide Crypto-Zusätze rechnen "$c = m^k \bmod n$" mit sehr großen Zahlen (1024Bit oder mehr). Diese Berechnung sollte aufgrund ihres sehr schnellen Multiplikationsvermögens in den DSPs durchgeführt werden. Noch besser wäre es, wenn die Chipkarte die RSA-Berechnungen durchführen würde, da der Secret-Key in ihr einen guten Schutz fände. Aber solche freiprogrammierbaren RSA-Chipkarten stehen den privaten Anwendern noch nicht zur Verfügung. Soll ein RSA-Algorithmus implementiert werden, muß zusätzlich ein Protokoll zwischen den beiden Crypto-Telefonen laufen, das den verschlüsselten Sitzungsschlüssel und die Benutzerdaten überträgt.

7 Anhang

7.1 Abbildungsverzeichnis

Abbildungsverzeichnis

7.2 Index

7.3 Glossar

7.4 Liste der Signalnamen

7.5 Pinbelegung der Steckverbinder

7.6 Stücklisten

7.6.1 Stückliste ISDN-Telefon-Board

7.6.2 Stückliste ISDN-DSP-Verschlüsselungsboard

7.6.3 Stückliste der externen Bauteile

7.7 Inhalt der Daten-CD

7.8 Literaturverzeichnis

Literatur

[1] "Applied Cryptography" second edition von Bruce Schneier
beim Wiley-Verlag ISBN 0-471-11709-9

[2] "Technik der Netze" 3. Auflage von Gerd Siegmund
beim R.v.Deker-Verlag ISBN 3-7685-2495-7

[3] "Das Mikrocontroller Kochbuch" von Andreas Roth
beim IWT-Verlag ISBN 3-88322-225-9

[4] "Handbuch der Chipkarten" von Rankl / Effing
beim Hanser-Verlag ISBN 3-446-17993-3

[5] PGP-Verschlüsselungssoftware (PGP v2.6.2i) von Philip Zimmermann *

[6] Datenbuch "TMS320C2x" von Texas Instruments *

[7] Datenbuch "AM79C30A" von AMD *

* mit auf der Daten-CD

7.9 DSP-Listing

7.10 Stromlaufpläne

7.11 Bestückunspläne

7.12 Layouts