CryptCard
Das Sicherheitssystems für Notebooks (Entwicklung 1992-1994)

Eberhard von Faber
September 2018

In den 1980er Jahren fand eine zunehmende Dezentralisierung der Datenverarbeitung statt. PCs hatten sich längst etabliert. Anfang der 1990er Jahre wurden die PCs sogar mobil. Diese Entwicklungen führten zu einem erhöhten Sicherheitsbedarf (Schutz der Daten), der überwiegend durch Software gedeckt wurde, die die Daten z.B. durch Verschlüsselung vor unberechtigten Zugriff schützten. Nachteilig wirkte sich dabei aus, dass der Computer langsamer wurde. Die CE Infosys GmbH in Bodenheim war eine der Firmen, die hardware-basierte Sicherheitssysteme entwickelte und anbot, die diesen Nachteil wettmachten und höhere Performance und höhere Sicherheit boten. CryptCard war das erste hardware-basierte Sicherheitssystem für Notebook-Computer. Das Konzept, die Elektronik, die Software im Innern und das Gehäuse wurden vollständig von mir entwickelt.

1 Übersicht

Insbesondere bei mobilen Computern besteht die Gefahr, dass das Gerät verloren geht oder gestohlen wird. Auch wenn sich der Computer aufgrund des Passwortschutzes nicht einfach starten lässt, kann der Finder bzw. Dieb sehr einfach die Festplatte ausbauen und die Daten mit einem anderen Gerät auslesen. Die 4 mm dicke, ca. 33 Gramm schwere CryptCard fungiert als *Ausweismedium*, um den Computer überhaupt starten zu können, und sie übernimmt mit ihrer Hochleistungselektronik die *Verschlüsselung* aller Daten in Höchstgeschwindigkeit und „im Fluge“ ganz ohne Nutzerinteraktion. Und da das System ohnehin tief in die Innereien des zu schützenden Notebook-Computers eingreift, sind auch gleich noch Sicherheitsfunktionen realisiert, wie eine *Verwaltung der Nutzer und ihrer Rechte*. Es werden verschiedene Verschlüsselungsmodi unterstützt, man kann Schnittstellen sperren u.v.a.m. CryptCard hat eine *Logbuchfunktion* mit interner Echtzeituhr. Abb. 1 zeigt ein Zielsystem mit der CryptCard.

Abb. 1: Das Flaggschiff T4700CT mit der Erweiterung CryptCard (nicht eingesteckt)

Die CryptCard[1] ist ein vollständiger, unabhängiger und autarker Microcomputer im Scheckkartenformat (siehe Abb. 2). Damit kann die Karte ihre Funktion als Sicherheitssystem optimal wahrnehmen. Links neben der Batterie erkennt man die CPU (8-Bit Prozessor) mit eingebautem PROM. Die beiden länglichen Toshiba-Chips sind batteriegepufferte Speicher (RAM) mit zusammen 256 KByte. Dort werden die umfangreichen Programme und die kryptografischen Parameter (Prüfwerte, Schlüssel usw.) gespeichert. Der große Chip (144 pins) in der Mitte ist der damals schnellste DES-Prozessor, der große Chip daneben enthält nahezu die gesamte Logik, da für andere Chips schlichtweg kein Platz war. Unten rechts sieht man die Echtzeituhr (batteriebetrieben).

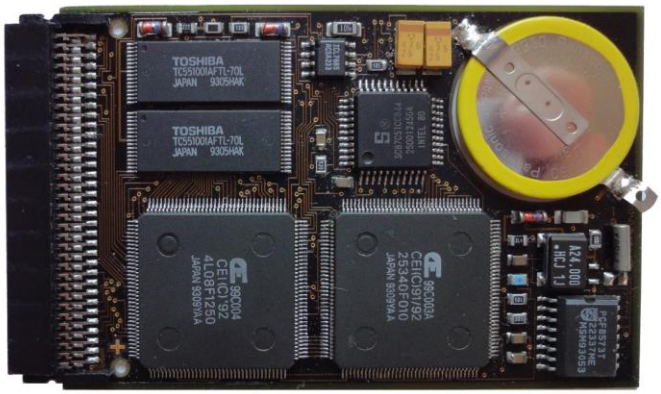

Abb. 2: Innenleben der CryptCard

Es wurden zwei Versionen der CryptCard entwickelt. 1) Die Toshiba-Version ist direkt mit der Elektronik bestimmter Toshiba-Computer verbunden und dort ins BIOS integriert. Nach der Installation ist das Notebook ohne die CryptCard wertlos und kann nicht betrieben werden. 2) Die Infosys-Variante kann für Notebooks verschiedener Hersteller genutzt werden. Diese Variante ist in Abb. 3 zu sehen.

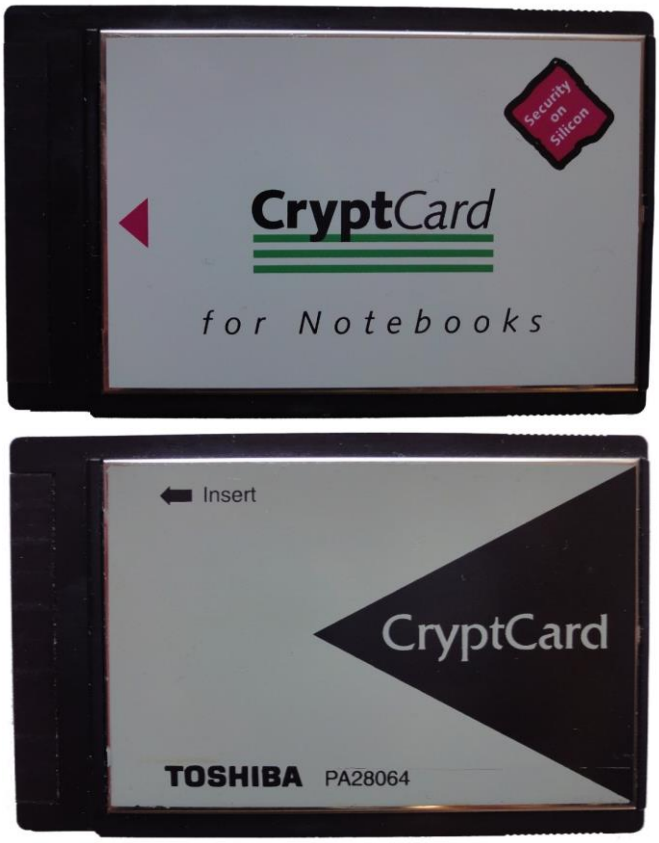

Abb. 3: Vorderseite der CryptCard (oben: Infosys-Version, unten: Toshiba-Version)

[1] Wer heute im Netz nach CryptCard sucht, findet eine Geschenkkarte, die NICHTS mit dieser CryptCard zu tun hat.

Tab. 1 enthält ein Profil und verschiedene technische Daten des CryptCard-Sicherheitssystems mit der CryptCard als Zugangsmedium und Sicherheitsmodul.

Tab. 1: Profil und technische Daten CryptCard

Funktion	Sicherheitssystem für Notebooks: Boot-Schutz (Karte und Passwort), transparente Verschlüsselung von Festplatte und optional von Disketten, NIST Zertifikat, Nutzungskontrolle von Schnittstellen (serieller und parallele Ports) und Computer (zeitliche Einschränkung), Logbuch-Mechanismen, hierarchisches Administrations- und Zugriffskonzept für Anwendungen in größerer Organisationen, BSI zertifiziert für T4600 und T4700 (mit gleicher Funktion einsetzbar mit T4800); ebenfalls „ThinkPad Proven“ (IBM-Qualitätssiegel)
Veröffentlichung	März 1993 (Computermesse CeBIT)
Formfaktor	PCMCIA Typ II Karte (4 mm × 86 mm × 54 mm; Kreditkartengröße)
Platine	vier Lagen, Elektronik über SMD (Surface-Mount Devices)
Prozessor	8051 mit internem 4 KByte PROM für das Startprogramm und den Bootloader (87C51, 44 pin quad flat pack)
Speicher	128 KByte RAM batteriegepuffert für 8051-Programme und Daten, ohne Verbindung zum PCMCIA-Bus und nur intern nutzbar; 128 KByte RAM batteriegepuffert für X86-Programme und Daten, 8 Pages mit Verbindung zum PCMCIA-Bus, Programmspeicher schreibgeschützt; (2x 32 pin TSOP)
Schnittstelle	16 Bit Input/Output und 256 KByte Speicher über 68 pin Steckverbinder (PCMCIA)
ASIC 1 (Eigenentwicklung)	SuperCrypt: weltweit schnellster DES-Prozessor (Verschlüsselungschip für kryptgrafische Operationen nach dem Data-Encryption-Standard, DES; NIST zertifiziert; 16 Mbyte/s) (kundenindividuell, 144 pin quad flat pack)
ASIC 2 (Eigenentwicklung)	CryptCard-Logik mit 64 Byte dual-interface-FIFO I/O-Speicher; realisiert den PCMCIA-Standard und verbindet alle anderen ICs untereinander und mit der externen PCMCIA-Schnittstelle (68 pins) (kundenindividuell, 144 pin quad flat pack)
Echtzeituhr, batteriegepuffert	RTC ohne Jahr (Jahr sowie Schaltjahrkorrektur usw. per Software) (16 pin quad flat pack)
Takt	24 MHz Quarz für Takt plus 32,678 kHz Uhrenquarz; 33 MHz PCMCIA-Bustakt (vom Notebook)
Stromversorgung	3 V Lithium-Batterie bzw. 5 V PCMCIA-Bus, Umschaltung und Spannungsüberwachung durch ICL7665
Anzahl der ICs	7
Anzahl passiver Bauelemente	25 (einschl. einer Batterie und zwei Quarzen)

2 Toshiba-Projekt und Infosys-Version

Es stellt kein Problem dar, dass die meisten sicherheitsrelevanten Daten in prinzipiell mit mechanischer Gewalt zugänglichen RAM-Bausteinen gespeichert sind. Zum einem können sie mit einem kryprografischen Schlüssel verschlüsselt werden, der sich in den Tiefen des in der CPU befindlichen PROMs befindet. Dies erhöht aber nur den Aufwand für einen möglichen Angreifer.[2] Der weitaus gewichtigere Grund ist der, dass die CryptCard ein Ausweismedium darstellt und sich (wie eine Chipkarte) in Hand des Nutzers befindet. Der hat ohnehin auf die Karte achtzugeben, da sich sein Computer ohne diese Karte nicht starten lässt. Dies ist bei der Toshiba-Version 100%-ig ernst gemeint. Einmal installiert, ist das CryptCard-Sicherheitssystem mit der Hauptplatine des T4700CT verbunden. Nicht einmal der Austausch der Festplatte ändert etwas daran. Ohne die richtige CryptCard ist der Computer wertlos. Nur ein autorisierter Nutzer kann diese Verbindung durch Deinstallation wieder lösen, wofür er jedoch eine mit den Systemparametern ausgestattete CryptCard braucht.

Vorbereitet wurde diese Integration mit Hilfe eines Vorgängers des T4700CT. Der T3300SL besaß eine 386er Low-Power-CPU und war mit 2 MByte Hauptspeicher ausgerüstet, der mit Hilfe eines Speicher-Erweiterungs-slots auf bis zu 18 MByte aufgerüstet werden konnte. Die eigentliche Neuheit war aber die 16-Bit-Erweiterung, die scheckkartengroße PCMCIA-Karten aufnehmen konnte wie z.B. die gerade in der Entwicklung befindliche CryptCard. Toshiba (Europa), Neuss, und die CE Infosys GmbH, Bodenheim, verabredeten eine Kooperation. Die ersten Tests wurden mit einem T3300SL durchgeführt, den man in Abb. 4 sieht. Ich habe den PCMCIA-Schacht aufgesägt, um das erste Evaluierungsmuster der CryptCard-Platine einstecken zu können, die jedoch nicht mehr erhalten ist. Die Chips und andere Bauelemente waren viel zu hoch, um im engen Typ-II-Schacht unterkommen zu können. Bei den ersten Tests im Herbst 1992 stellt sich in Tokio allerdings heraus, dass der 16-bit I/O-Transfer nicht funktioniert. Der Notebook-Hersteller hatte die PCMCIA-Spezifikation falsch interpretiert. Der Chip mit nach meiner Erinnerung 386 Pins hatte einen Fehler. Glücklicherweise hatte ich meine inoffizielle Kopie richtig gelesen. Mein viel kleinerer Chip hatte bestanden. Das wurde dann nach einigem Hin und Her auch akzeptiert. Das waren bange Minuten. Es dauerte, bis die Anspannung nachließ, schließlich ging es um ein wichtiges Projekt für die Infosys und mich.

Abb. 4: T3300SL mit der CryptCard im aufgesägten PCMCIA-Schacht

[2] Zusätzlich ist es herstellerseitig möglich, den SuperCrypt-Chip batteriegepuffert zu betreiben. Dieser besitzt write-once/read-never Speicher für Masterkeys. Die bereits erwähnte Lösung der Speicherung im Speicher der internen CPU setzt eine Bestückung mit dem 83C851 anstelle des 87C51 voraus.

Die wichtigsten Tests konnten durchgeführt und wichtige Schnittstellen spezifiziert werden. Aus dem ersten Evaluierungsmuster wurde ein Prototyp in Originalgröße. Zur Weiterentwicklung und weitere Tests fuhr ich wieder nach Tokio. Da es technische Schwierigkeiten gab, wurde der Aufenthalt mehrfach verlängert. Indes ging mir das Geld aus, und ich hatte kaum mehr etwas zum Anziehen, denn zum Waschen hatte ich keine Zeit und das Hotel war nicht darauf eingerichtet. Tag und Nacht wurde programmiert. Auch habe ich ca. dreimal das Hotel wechseln müssen, da die Buchungen ausliefen. Ein freies Hotel war schwierig zu finden, weil der Tokio-Marathon anstand. So wurde ich Gast in einem Ryokan, einem traditionell eingerichteten und betriebenen Hotel. Eine interessante Erfahrung.

Wie bereits angedeutet, hatte auch dieser Aufenthalt sein Desaster. Diesmal allein für mich. Die Zusammenarbeit stand kurz vor dem Abbruch und konnte offenbar nur durch aus Deutschland eingeflogene Manager abgewendet werden, die die Bedeutung des Sicherheitssystems für den Markterfolg in Europa herausstellten. Was war geschehen? Da ich nicht ins Allerheiligste (die Entwicklungsabteilung) durfte, war ich in einem separaten Raum untergebracht. Der Austausch erfolgte durch gemeinsame Treffen und durch das Hin- und Hertragen der jeweils neu programmierten CryptCard, denn nur ich hatte eine Entwicklungsumgebung dafür. Nun passierte es häufig, dass die CryptCard allzu früh mit der landesspezifisch ausgesprochenen Bemerkung „RAM-Code lost“ zurückgebracht wurde. Dies konnte, das war mir schnell klar, nur durch eine Unterbrechung der Spannungsversorgung verursacht worden sein. Aber es fehlten mir die Mittel, den Fehler zu finden. Immerhin erhielt ich eine weitere Chance. Zurück in Bodenheim war der Fehler schnell gefunden. Der für die Umschaltung der Spannungsversorgung verantwortliche Minischaltkreis hatte in seinen Ausgangsstufen einen anderen Transistortyp verbaut als angeben bzw. als ich aus den internen Stromlaufplänen herausgelesen hatte. Da ich aus Platzgründen um jedes Bauelement auf der Platine feilschen musste, hatte ich wohl etwas überreizt. Der nur ab und zu auftretende Fehler war am heimischen Arbeitsplatz nicht aufgefallen, da die Karte meist fremdversorgt wurde. Die notwendige Änderung der Schaltung war schwierig zu realisieren, weil eigentlich kein Platz für zusätzliche Bauelemente war. Doch dazu mehr im nächsten Kapitel. Zuvor noch etwas Technik zu den beiden Versionen.

Die Toshiba-Version ist die ursprüngliche. Hierbei muss der Computer vor dem Starten des Betriebssystems die CryptCard erkennen. Sodann läuft ein (kryptografisches) Protokoll, bei dem Computer und CryptCard jeweils prüfen, ob sie zueinander gehören. Andernfalls verstummt der Computer und die CryptCard bietet keine weiteren Dienste an. Der Computer braucht die CryptCard, um Daten von der verschlüsselten Festplatte (oder Diskette) lesen zu können. Die CryptCard bietet die Dienste zur Entschlüsselung nur dann an, wenn sie im richtigen Computer steckt. Erkennen sich beide als einander zugehörig, kann es weitergehen. Doch bevor das Betriebssystem von der verschlüsselten Festplatte gestartet werden kann, muss der Nutzer noch beweisen, dass er der berechtigte Besitzer der CryptCard ist. Das erfolgt durch Eingabe des richtigen Passwortes.[3] Danach ist das Sicherheitssystem unsichtbar für den Nutzer, es sei denn, er versucht eine Funktion zu nutzen, die sein Administrator gesperrt hat. Dies gibt es natürlich aber nur bei Anwendungen in größeren Organisationen.

Für die Toshiba-Version bot der Computer-Hersteller einen Aufkleber an, der auf dem Computer angebracht, potenzielle Diebe abschrecken sollte (links in Abb. 5 zu sehen). Gleichzeitig war damit wohl auch Werbung verbunden. Die Infosys-Version, die gleich erklärt wird, sollte mit kleinen Pappkärtchen ins Gespräch gebracht werden (rechts in Abb. 5).

[3] CryptCard implementiert konfigurierbare Passwortrichtlinien und beschränkt die Anzahl der Falscheingaben, setzt die Gültigkeitsdauer bzw. den Passwortwechsel durch und achtet darauf, dass Nutzer nicht alte Passworte erneut verwenden.

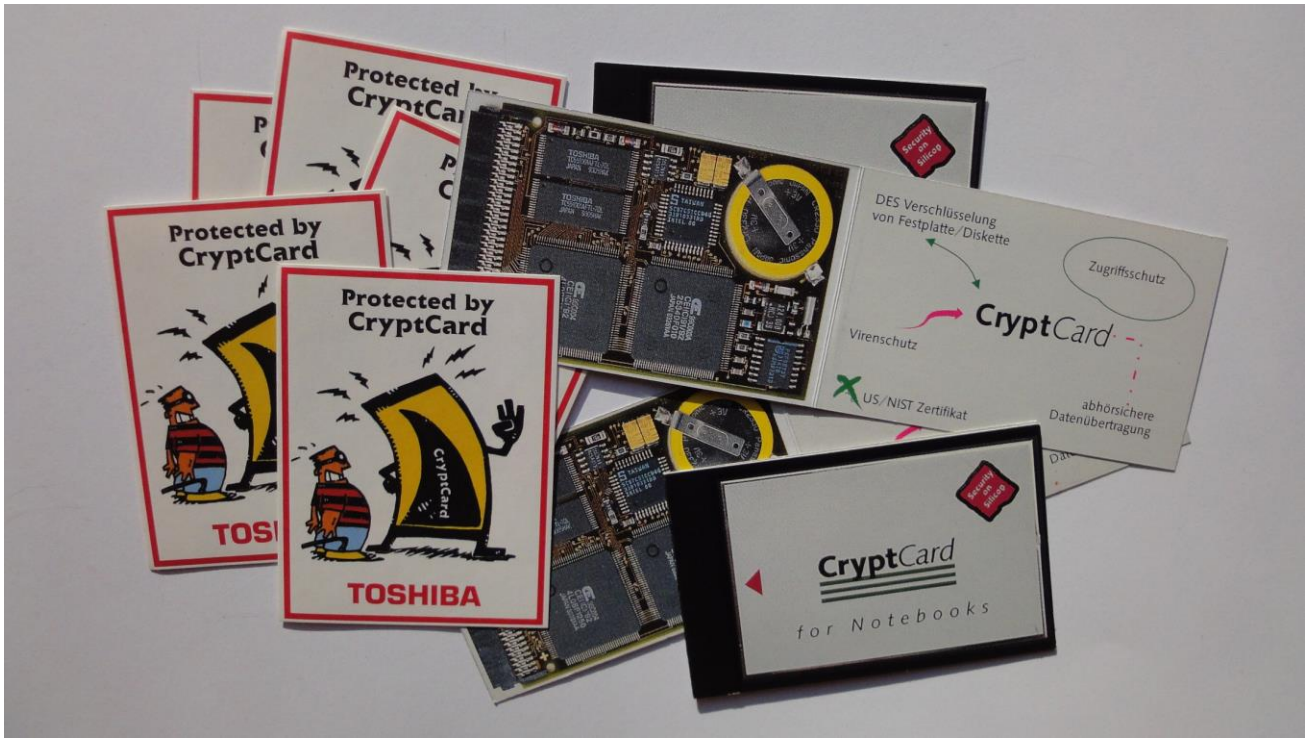

Abb. 5: Aufkleber der Toshiba-Version und Pappkärtchen für die Infosys-Version

Die Infosys-Version entstand später. Andere Notebooks von Toshiba und anderen Herstellern besaßen nicht die speziellen Anpassungen für die CryptCard. Um die CryptCard trotzdem nutzen zu können, wurde sie wie viele herkömmliche Sicherheitssystem integriert. Wenn nichts anderes eingestellt ist, liest ein PC als erstes einige bestimmte Sektoren der Festplatte und führt das darin enthaltene Programm aus. Die Sektoren bilden den Master-Boot-Record (bzw. Partition Sector), der wiederum andere Sektoren nachlädt, so dass schließlich das Betriebssystem gestartet wird. Das CryptCard-Sicherheitssystem modifiziert bei der Installation den Master-Boot-Record (bzw. Partition Sector) so, dass ein CryptCard-spezifisches Programm ausgeführt wird. Es überprüft, ob die CryptCard im richtigen Rechner steckt und prüft das Passwort des Nutzers. Ist beides korrekt, bietet die Karte ihre Entschlüsselungsfunktionen an, und nun wird der originale Master-Boot-Record (bzw. Partition Sector) geladen und ausgeführt, und das Betriebssystem wird von der verschlüsselten Festplatte gestartet. Wieder bemerkt der Nutzer im Folgenden nichts, es sei denn, er will eine Funktion nutzen, die durch das System gesperrt ist.

Führende Hersteller von Notebooks wollten ihre Geräte mit dem CryptCard- Sicherheitssystem anbieten und waren an unserer „Freigabe“ sehr interessiert. Denn Technik ist nur prinzipiell kompatibel, und man tut gut daran, die Funktion durch Tests zu überprüfen. Deshalb hatte ich ein Testlabor aufgebaut. Es gab sogar ein Bottom „CryptCard approved“. Doch das wurde nicht immer verliehen, was eine glatte Untertreibung ist. Die neuesten Modelle fielen reihenweise durch. Hersteller hatten PCMCIA in ihre Spitzenmodelle für Manager und andere Vielreisende integriert, damit ihre Notebooks mit mehr Mobilität ausgestattet werden konnten. Die genügsamen Modems und Speicherkarten funktionierten auch prima. Nur die auf maximalen Datendurchsatz angewiesene CryptCard machte häufig Probleme. Aber die Ursache war immer das Notebook, auch wenn mancher Hersteller die Schuld anfangs woanders zu suchen versuchten. Die Inkompatibilität enttäuschte die Hersteller der Computer und den der CryptCard gleichermaßen. Schließlich war die CryptCard zum Statussymbol avanciert und sehr bekannt geworden, auch wenn sie das Schicksal vieler Sicherheitssysteme teilte: Man redete viel darüber, fand aber oftmals Gründe, sie lieber nicht einzusetzen.

3 Platine und Logik-ASIC

CryptCard war nicht das erste Sicherheitsprodukt der Firma Computer Elektronik Infosys, im Gegenteil. Star der Firma war „ELKEY“, eine sperrige Platine, die man in einem Desktop-PC montieren konnte. Das Projekt „CryptCard“ begann eigentlich damit, dass der Geschäftsführer mich unbedingt einstellen wollte, weil ich als Student selbst einen Computer namens Lotus 2000 entwickelt und aus Einzelteilen zusammengebaut hatte.

Auch das „Betriebssystem“ hatte ich geschrieben. Nachdem ich zum Jahresbeginn 1992 angeheuert hatte, gab mir der Geschäftsführer und Inhaber, Georg Krause folgenden Auftrag: „Machen Sie mir die ELKEY so klein, dass sie in einem der neuen Notebooks Platz findet.“ Dazu legte er mir einen dicken Leitz-Ordner auf den Schreibtisch, der die Kopie des PCMCIA-Standards enthielt, in dem der kleine Erweiterungsschacht spezifiziert war.[4] Das war bis zur Fertigstellung des ersten Prototyps fast der einzige Außenkontakt für mich.

Normalerweise geht man so vor: Man entwickelt eine Schaltung, auf Basis vorhandener Chips und anderer Bauelemente. Falls es keine passenden Standardschaltkreise gibt, was selten der Fall ist, entwirft man kundenindividuelle Chips (ASICs) und lässt diese produzieren. Nun wird die Leiterkarte entworfen, die alle Chips und die anderen Bauelemente miteinander verbindet. Bei CryptCard war dies anders. Die Leiterkarte bot wenig Platz (siehe Abb. 6). Da sie wegen der Dicke nur vier Lagen haben sollte (zwei für Signale, zwei für die Spannungsversorgung) und die sieben ICs mit insgesamt 420 Pins und der Steckverbinder mit 68 Pins zu Buche schlugen, war die Leitungsführung auf der Platine Schwerstarbeit.[5] Immer wieder wurde die Signalbelegung der Pins des Logik-ASICs in der Datenbank neu gewählt. Dann wurde ein neuer Versuch gestartet, die Chips und die Bauelemente miteinander zu verbinden. Und natürlich wurden zwischendurch auch immer mal die Bauelemente verschoben. Einige Pins des Logik-ASICs wurden nur dafür genutzt, um eine Verbindung vom einen Ende zum anderen zu schaffen, ohne dass dies durch die Platine erfolgte. Aber das war die Ausnahme, zeigt aber die Not, in der wir steckten. Am Ende schien alles zu passen. Nur ein winziger Widerstand oder Kondensator passte einfach nicht. Da musste ich die Schaltung nochmals ändern, um eines der passiven Bauelemente einsparen zu können. Erst jetzt konnte der Logik-ASIC in Auftrag gegeben werden.

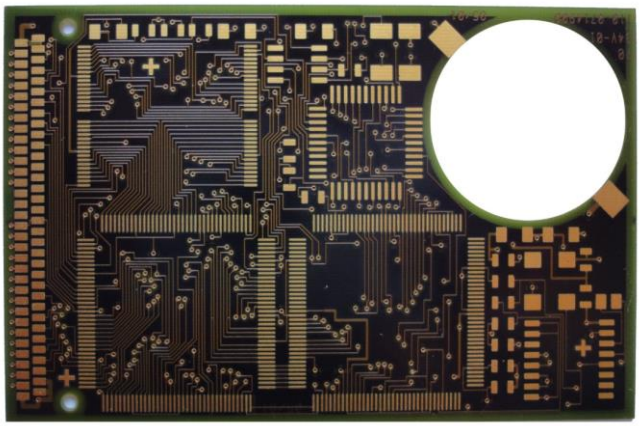

Abb. 6: Platine der Karte (unbestückt)

Die Entwicklung des Logik-ASICs war natürlich weitaus anspruchsvoller als die der Leiterkarte. Er musste die komplexen Funktionalitäten der PCMCIA-Schnittstelle bereitstellen, den DES-Prozessor SuperCrypt für externe Aufgaben (x86-CPU des Notebooks) sowie interne (8051-CPU auf der Karte) einsetzbar machen, die RAM-Speicher für externe und interne Nutzung bereitstellen und einen Briefkasten („Command Exchange Unit“) bereitstellen, damit die x86-CPU des Notebooks bzw. seine Software und die 8051-CPU auf der Karte miteinander kommunizieren konnten. Jede dieser Aufgaben musste durch feststehende Elektronik (und nicht durch wandelbare Software) umgesetzt werden. Ich werde Beispiele dafür bringen, was der Schaltkreis leisten musste.

Der Briefkasten („Command Exchange Unit“) war ein kompliziertes Teil, u.a. weil ich es aus einfachen Standard-Superzellen zusammensetzen musste. Das waren die vordefinierten Bausteine, die der Chiphersteller (in unserem Fall ebenfalls Toshiba) überhaupt ins Silizium implantieren konnte. Überträgt das Notebook (Host in Abb. 7) Daten an die CryptCard (MC), so füllt sich der 64 Byte große Briefkasten von oben nach unten. Liest die Karte diese Daten, so leert er sich. Ähnlich funktioniert die umgekehrte Kommunikation von der Karte (MC) zum Notebook (Host), rechts im Bild. Statussignale zeigen an, wenn der Puffer voll bzw. leer ist bzw. wenn

---

[4] Im Januar erarbeitete ich jedoch zuerst eine Lösung, die vollständig auf der Hauptplatine eines Toshiba-Notebooks implementiert werden sollte. Das Projekt wurde nicht realisiert. Offenbar tauchte etwa im Februar oder Anfang März der PCMCIA-Standard bei Herrn Krause auf und das Projekt nahm seine Wendung.

[5] Das Routing auf der Platine habe ich nicht selbst durchgeführt. Ich habe mir nur die Probleme erklären lassen und dann Lösungsideen erarbeitet, die dann ausprobiert wurden.

Daten zum Abholen bereitstehen oder abgeholt wurden. Die wirklich komplizierte Logik des gesamten Schaltkreises, der viel mehr als diesen einfachen Briefkasten implementierte, hatte einen einzigen Fehler, den ich erst bemerkte, als ich die in einigen Wochen für viel Geld produzierten Chips ausprobieren konnte. Der Fehler lag in diesem Briefkasten (64 Byte dual-interface-FIFO I/O-Speicher, speziell entwickelt für diese CryptCard). Die Kosten für eine erneute Herstellung waren enorm. – Aber ich fand einen Weg, den Fehler durch eine Änderung der Software auszugleichen.

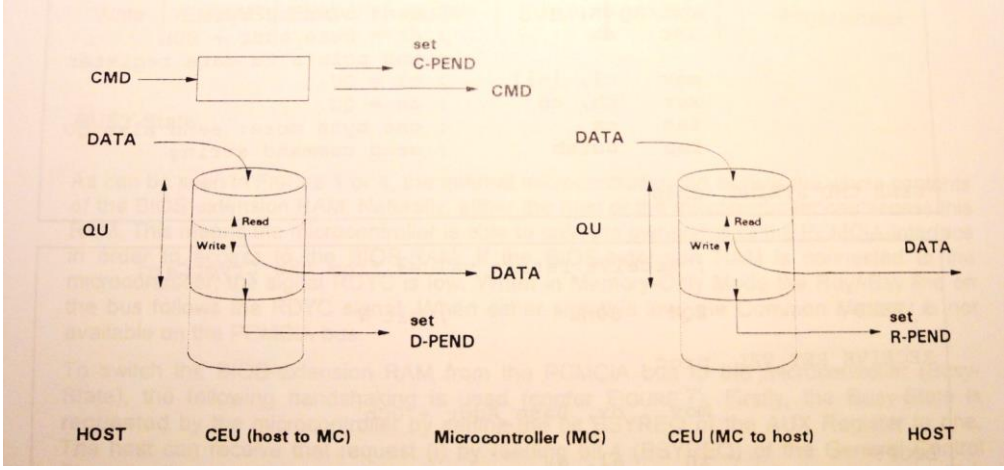

Abb. 7: Veranschaulichung der Command Exchange Unit

Die Verwaltung der beiden RAM-Speicher und des SuperCrypt-Chips ist ebenfalls komplexer als man denkt. Einer der beiden RAMs kann sowohl von der x86-CPU des Notebooks als auch von der internen 8051-CPU der CryptCard verwendet werden. Das gleiche gilt für Datenports des SuperCrypt.

Dass der SuperCrypt-ASIC von beiden Seiten gebraucht wird, ist klar. Das Notebook erledigt mit ihm die Verschlüsselung und Entschlüsselung der Festplatte (und evtl. von Disketten) über die schnelle PCMCIA-Schnittstelle (16-Bit-I/O). Die interne CPU benötigt den Zugriff auf den Chiffrierchip z.B. für Key-Management-Aufgaben. Der Logik-ASIC ermöglicht die entsprechende Umschaltung durch Programme der internen 8051-CPU. Der SuperCrypt wird mit 24 MHz getaktet.

Doch beim Hochfahren des Notebooks gibt es zunächst gar keinen I/O-Kanal zur Karte. Beim Start des Rechners erwartet dieser 4 KByte sogenannten „Attribute Memory“ mit der „Card Information Structure (CIS)“ sowie einige Signal- und Steuerregister, um die PCMCIA-Karte korrekt einbinden und ansteuern zu können.

Im Normalbetrieb werden 32 KByte belegt, die vom Logik-ASIC ganz anders gefüllt werden. 16 KByte werden durch Speicher belegt, die Programme für die x86-CPU enthalten. Hier blendet der Logik-ASIC je nach den Vorgaben des internen Programms einen von sieben 16 Kbyte großen Speicherblöcken ein, die in der Regel permanent schreibgeschützt sind. Dazu kommen 12 KByte RAM. Dazu kommen Register zur Nutzung der CryptCard, die der Logik-ASIC als „Memory-Mapped-I/O“ einblendet, die jedoch verschwinden, wenn die Karte in den I/O-Modus geschaltet wird. Es werden also 32 KByte belegt.

Sowohl bei der Toshiba-Version als auch bei der Infosys-Version befindet sich die Software, die für die Ver- und Entschlüsselung von Festplatte und Daten sorgt und weitere Sicherheitsfunktionen bereitstellt, auf der CryptCard in den sieben 16 KByte Speicherblöcken. Ein Beispiel für diese weiteren Sicherheitsfunktionen ist die Präsentation des Anmeldebildschirms mit der Abfrage und Prüfung des Passwortes des Nutzers. Geschrieben wird der Speicher durch den 8051-Microcontroller, der den Logik-ASIC so umschalten kann, dass die Speichersegmente für ihn sichtbar sind. Dazu müssen Adress- und Datenbus sowie Steuerleitungen umgelegt werden. Die 7 x 16 KByte RAM sind batteriegepuffert, so dass die Programme nicht verloren gehen. Diese Konstruktion schließt Manipulationen der Software aus. Sie ist schreibgeschützt und kann nur über Kommandos der CryptCard geladen bzw. aktualisiert werden, wobei die internen Programm der 8051-CPU die Echtheit und Unversehrtheit des Codes prüfen. Dies gilt für alle derartigen Veränderungen einschließlich der Einstellungen der CryptCard.

Zu Schluss noch ein kurzer Blick darauf, wie der Logik-ASIC und der CryptCard-ASIC überhaupt entwickelt wurden. Ein ASIC ist kein programmierbarer Schaltkreis. Er wird kundenindividuell in einer Chipfabrik

hergestellt. Was der Schaltkreis macht und wie er sich verhält wird vollständig durch den Entwickler bestimmt. Das war ich in diesem Falle. Im Prinzip handelt es sich zunächst um eine Entwicklung einer digitalen Schaltung, die jedoch keine passiven Bauelemente enthalten kann. Anstelle der normalerweise eingesetzten Integrierten Schaltkreise (Chips) kommen sogenannte Standard-Zellen zum Einsatz, die der (zukünftige) Hersteller des ASIC definiert und in einer Bibliothek mit sämtlichen Parametern abgelegt hat. Die Schaltung wird mit einem CAD-Programm entwickelt, bei der man Standard-Zellen lädt und mit der Maus verbindet. Diese CAD-Software lief unter DOS, war aber graphisch – auch schon 1992. Das Schöne an der CAD-Software ist, dass man Tests machen kann. Man kann die Schaltung also ausprobieren. Das hat aber nur dann einen Wert, wenn man genau weiß, was man erreichen will oder muss. Bei der Umsetzung des PCMCIA-Standards war das oft nicht einfach. Aber auch das Konzipieren nimmt einem das Programm nicht ab. Man muss selbst wissen, was man will. Stimmt alles nach Einschätzung des Entwicklers, so ist die Arbeit längst nicht getan. Da der Fertigungsprozess Toleranzen hat, muss überprüft werden, ob die Schaltung auch unter allen möglichen Bedingungen fehlerfrei funktioniert. Das wird mit einem Testprogramm des Herstellers getestet, das die Abweichungen bzw. Toleranzen berücksichtigt. Jetzt wird der ASIC nur noch von außen betrachtet. Der Entwickler muss für jeden Takt für jeden der 144 Anschlüsse des Chips bestimmen, ob es sich um einen Ausgang handelt und ob der die Zustände High, Low oder Floating annimmt, oder ob dies ein Eingang ist und ob der mit Low oder High gefüttert wird. Damals waren diese Informationen als reine Textdatei zu erzeugen. D.h., ich musste eine Textdatei erzeugen, die 144 „Spalten“ bzw. Werte in einer Zeile hatte, die die Ausgaben und Eingaben festlegten. In der Vertikalen war das Ganze sehr viel länger, denn die Werte mussten für jeden Takt festgelegt werden. Das klingt nicht besonders schwer. Aber wenn man bedenkt, dass man z.B. die „Command Exchange Unit“ mit 64 Byte füttern und auslesen musste und jedes Schreiben und Lesen jeweils mehrere Takte umfasste, dann ahnt man, wie lang die virtuelle Tapetenrolle war. Die Forderung des Herstellers war: Jede Leitung muss jeden möglichen Zustand angenommen haben. Das ist mehr, als man ahnt. Am Ende war alles fertig. Ich lieferte die Dateien stolz und mit etwas Bange ab. Und? Was kam zurück? Es gibt zwei oder drei Leitungen, die nicht alle Zustände angenommen haben. Ich musste nacharbeiten.

Ich kann es nicht beschwören, aber ich glaube, dass vom Beginn des Schaltungsentwurfs bis zur Ablieferung der finalen Unterlagen an den Hersteller gerade vier Monate vergingen. Und ich glaube, ich hatte in dieser Zeit noch ein paar andere Beschäftigungen.

4 Gehäuse und Fertigung

Es war leider nicht so, dass es die richtigen Gehäuse für die PCMCIA-Karte fertig zu kaufen gab. Ich war schon froh, dass die Steckverbinder (von Fujitsu) den Anforderungen entsprachen. Also musste ich das Gehäuse selbst entwerfen. Micrografx Designer war ein Windows-Programm, mit dem man technische Zeichnungen maßstabsgetreu anfertigen konnte. Und so ging es los. Zuerst wurden die äußeren Abmessungen aus der PCMCIA-Spezifikation entnommen und so angepasst, wie ich es brauchte. Die CryptCard ist nur auf der einen Seite dick (Typ II), auf der anderen Seite ist sie flach (wie Typ I). Siehe Abb. 8.

Der mechanische Aufbau war nicht ganz einfach. In der Zeichnung sind auch die Toleranzen eingetragen. Die Karte musste exakt geführt werden. War sie zu groß, blieb sie stecken oder konnte nur schwer entfernt werden. War sie zu klein, so bestand insbesondere die Gefahr, dass Stecker oder Steckverbinder beschädigt wurden. Da die Spritzform ebenfalls Toleranzen hat und der Plastikwerkstoff beim Hartwerden seine Form noch verändert, mussten Maße und Werkstoff genau aufeinander abgestimmt werden.

Die Toshiba-Version wurde zu einem richtigen Toshiba-Produkt und musste alle firmenüblichen Anforderungen erfüllen. Dazu gehörte, dass wir nachweisen mussten, dass CryptCard (und Notebook) die 30.000 spezifizierten Steckzyklen unbeschadet überstanden. Also haben wir in der Entwicklungsabteilung drei Arbeitsplätze eingerichtet, wo dies überprüft, also ausprobiert (getestet) wurde. Eine Software zählte und forderte zu einer kleinen Pause ein, falls umfangreichere Test anstanden. Denn wir wollten auch wissen, wann die ersten Fehler auftraten und einen wirklichen Nachweis für die volle Funktionsfähigkeit bis zum Punkt X haben. Um es vorwegzunehmen: Die Karten bestanden den Test. Die Durchführung war langwierig und nichts für einen Entwickler. Aber einige Mitarbeiterinnen aus der Produktion waren für etwas Abwechselung dankbar (jedenfalls anfangs) und dafür, in das Allerheiligste der Entwicklungsabteilung vordringen zu dürfen.

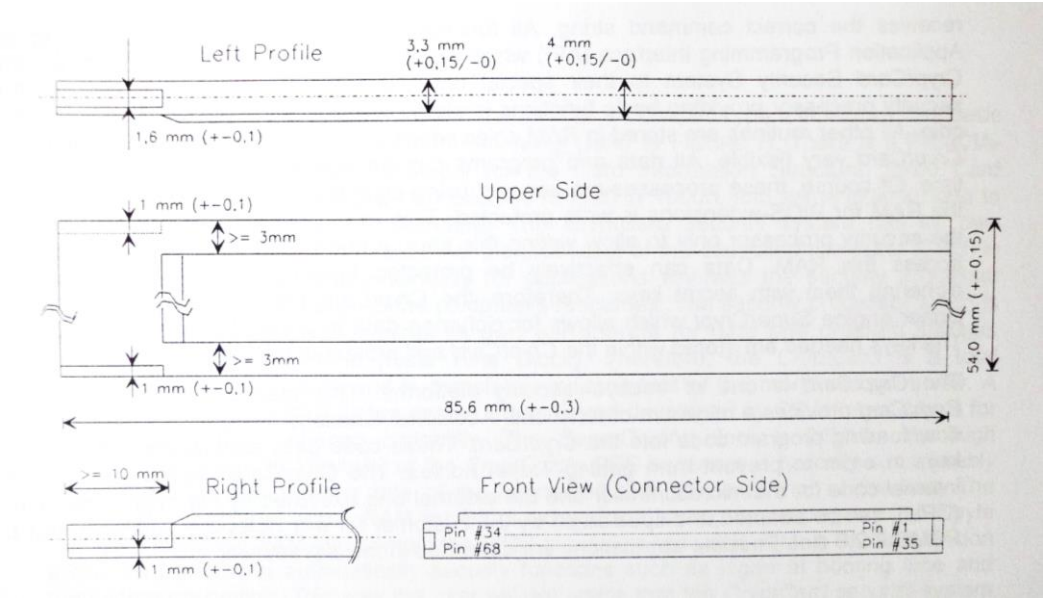

Abb. 8: Zeichnung für das Gehäuse (Micrografx Design)

In Abb. 9 (links) ist recht gut der Plastik-Rahmen zu sehen, in die der Steckverbinder mit der Platine eingesetzt wird.[6] Die obere Abdeckung (sie fehlt in der Abbildung) wird in den Rahmen eingerastet. Die untere Metallplatte (rechts in Abb. 9) liegt plan im Rahmen.

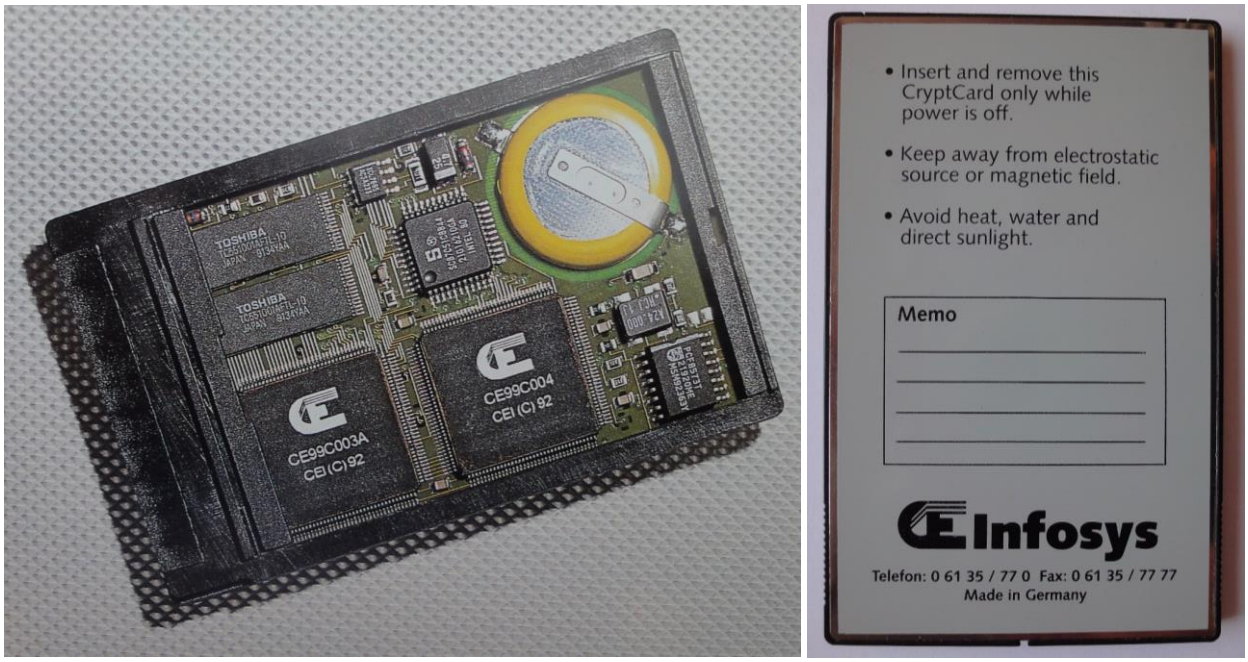

Abb. 9: Gehäuse von oben ohne Abdeckung (links) und von unten (rechts) geschlossen

Damit das Ganze zusammenhält, wird es verklebt. Und hier begannen die Probleme. Manch ein Kleber verzog die Karte beim Festwerden, andere führten zu Beschädigungen, und wir mussten lange probieren, um die richtige Menge von Kleber und den richtigen Andruck mit der richtigen Temperatur über die richtige Zeit zu finden. Das Verkleben war der letzte Vorgang bei der Fertigung der CryptCard vor der abschließenden Qualitätskontrolle und dem Verpacken. Davor musste die Elektronik natürlich erst aus die Leiterkarte.

Während der Entwicklung musste ich die ICs per Hand auf die Platine auflöten. Das fehlerfrei hinzubekommen ist nicht einfach, beträgt doch der Abstand zwischen den Pins der ASICs und RAMs nicht mehr als 0,3 mm. Die „Beinchen“ sind 0,2 mm breit. Erst als alles fertig und vollständig getestet war, konnte der neue SMD-Be-

[6] In diesem Bild für zahlreiche Flyer und Presseberichte sind die Beschriftungen der beiden ASICs gefälscht oder leider auch vertauscht worden.

stückungsautomat programmiert werden und seine Arbeit aufnehmen.[7] Dabei bestehen im Prinzip die gleichen Herausforderungen wie beim Löten per Hand. Die Bauteile müssen exakt positioniert werden und dürfen nicht verrutschen (weshalb sie festgeklebt werden), und die richtige Menge Lötzinn muss vorhanden sein, damit es bei der nachfolgenden Erhitzung das Bauteil mit der Platine verbindet.

5 Das fertige Produkt und seine Präsentation

CryptCard hat viele Sicherheitsfunktionen, insbesondere auch eine hierarchische Vergabe von Zugriffsrechten. Die Frage „Wer darf was?“ muss sich dabei an der Unternehmensorganisation bzw. der Hierarchie in einer Organisation orientieren. Es wurde daher angestrebt, dies bei der Verwaltung und Konfiguration der CryptCards abzubilden. Natürlich sind diese Funktionen für den Einzelnutzer uninteressant und werden dann nicht genutzt. Im Unternehmensumfeld besteht aber z.B. bereits die einfache Anforderung, dass jemand auf die Daten zugreifen kann, wenn der Besitzer von Karte und Rechner das Passwort vergessen hat oder dies vorgibt oder die CryptCard nicht mehr hat bzw. dies vorgibt. Und natürlich muss man verhindern, dass dadurch die Unternehmenshierarchie ausgehebelt wird und sich Mitarbeiter am unbeobachteten Notebook ihres Chefs zu schaffen machen. Deshalb hat das CryptCard-Sicherheitssystem ein hierarchisches Rechtemanagement, und es gibt Familien, die aus prinzipiell gleichberechtigten Nutzern bestehen. In einem Flyer von Toshiba sieht das so aus, wie in Abb. 10 zu sehen ist.

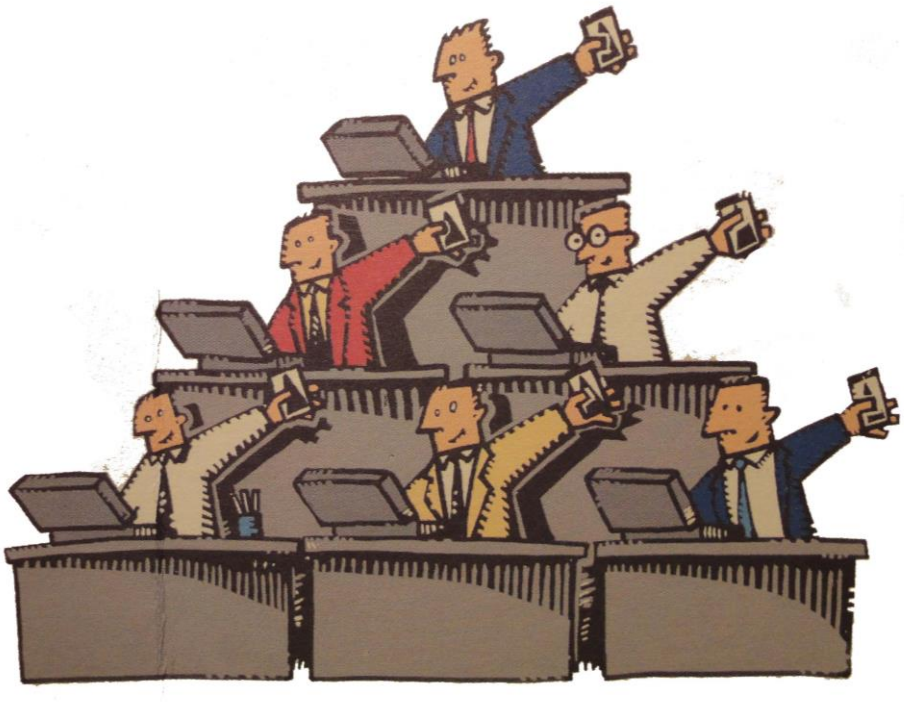

Abb. 10: Visualisierung der Rechteverwaltung in einem Toshiba-Flyer

Zu beiden Versionen (Toshiba und Infosys) gehörten neben der CryptCard natürlich auch ein Handbuch und eine Diskette (!) mit der Installationssoftware. Abb. 11 zeigt den T4700CT mit allen Heften, Flyern, Merkblättern, der Diskette und der CryptCard.

[7] SMD bedeutet Surface-Mount Device. D.h., es gibt keine Bohrungen in der Platine, sondern die Bauteile werden aufgelegt und dann verlötet.

Abb. 11: Der T4700CT mit allen Heften, Flyern, Merkblättern, der Diskette und der CryptCard

Zu weiteren Herausforderungen bei der Entwicklung gehörten das sichere Wiederherstellen defekter CryptCards, die Interoperabilität verschlüsselter Disketten, die sichere Aktualisierung von Software im Falle von Fehlern und die Unterstützung der Schlafmodi (Resume, Suspend), die heute z.B. „Energie sparen“ heißen und den Computer im Schnellstart zurück zum alten Arbeitsstand bringen. Eine der schwierigsten Aufgaben bestand jedoch darin, den Nutzern die vielfältigen Funktionen so zu erklären, dass sie diese gerne und fehlerfrei einsetzen. Auch heute ist dies noch das größte Problem vieler Sicherheitssysteme.

CryptCard wurde Anfang 1993 auf der CeBIT der Weltöffentlichkeit vorgestellt und mehr als zehn Jahre lang hergestellt und erfolgreich verkauft. Das System machte viele Notebooks zu den sichersten PCs der Welt.

Mein nächster Computer war sehr groß. Er war jedoch ziemlich dumm, aber dafür sehr schnell:
Diese Eigenschaft teilte er mit vielen Spezialisten. Er bestand aus sehr vielen,
gleichartig aufgebauten Teilen: Das hat ihn von anderen Rechners unterschieden.
Er hat auf eine ganz eigene, spezielle Weise für mehr Sicherheit gesorgt.
Und ein bisschen war er mit dem SuperCrypt verwandt.

Über den Erbauer

Spätestens nach Erreichen des Teenager-Alters habe ich angefangen, elektronische Geräte zu „basteln“. Anfangs half mein Vater mit Bauplänen, oder ich nutzte Bausätze. Doch schon bald stand die Entwicklung eigener Geräte im Vordergrund. Schon in der 7. Klasse war mir klar, dass ich Elektrotechnik studieren und dann in der Forschung und Entwicklung (F&E) arbeiten wollte. Allerdings hat mich auch die Physik interessiert, so dass ich beides studierte und danach auf dem Gebiet der Theoretischen Physik promovierte. Während meines Studiums baute ich mir einen Computer (Lotus 2000) aus Einzelteilen zusammen und programmierte auch das Betriebssystem. Das hat mir den Job bei CE Infosys verschafft, wo ich drei Jahre überwiegend am CryptCard-Sicherheitssystem arbeitete, wobei ich zuletzt vor allen Firmen bei dessen Einsatz half. Danach habe ich als Berater und Gutachter für IT-Sicherheit gearbeitet. Heute entwickle ich für einen großen IT-Konzern Methoden, Prozesse und Standards zur Absicherung einer großtechnischen, weit verteilten und komplexen IT-Produktion.

Prof. Dr. Eberhard von Faber

Literatur

[1] NIST Computer Security Resource Center: DES Validation List; April 1996, https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program/Validation/DES-(1977-1996)-Validation-List

[2] Toshiba Europe: Spec; http://www.toshiba-europe.com/computers/products/notebooks/t4700ct/product.shtm

[3] Eberhard von Faber: Datenschutz bei Notebooks; Design & Elektronik 4, 16.02.1993

[4] Ed Gray: CryptCard: Tight security for the mobile work force; fcw.com, Oct 03, 1999, https://fcw.com/articles/1999/10/03/cryptcard-tight-security-for-the-mobile-work-force.aspx

[5] Ariel Tam: CryptCard to protect notebook users; ZDNet, June 27, 2001, https://www.zdnet.com/article/cryptcard-to-protect-notebook-users/