Überblick zu Angriffe auf Chipkarten
von Wolfgang Rankl, München

11. April 2003

Der folgende Text ist eine zusammengefasste Version des Kapitels über Chipkartensicherheit im Handbuch der Chipkarten von Wolfgang Rankl und Wolfgang Effing, erschienen 2002 im Carl Hanser Verlag München.

Die Möglichkeit des Schutzes und der Geheimhaltung von Daten in Chipkarten ist einer der Hauptvorteile gegenüber allen anderen Datenträgern, wie Magnetstreifenkarten oder Disketten. Deshalb ist auch eine auf diesen Zweck abgestimmte und optimierte Chiphardware mit dazu passenden kryptografischen Verfahren zur Sicherung der geheimen Daten unabdingbar. Doch ist Sicherheit nicht alleine von der Spezialhardware des Mikrocontrollers oder den in der Betriebssystemsoftware realisierten kryptografischen Algorithmen abhängig. Auch die Sicherheit der Chipkarten-Anwendung und die Design-Prinzipien, die Entwickler auf dem Weg dorthin anwenden, sind von elementarer Bedeutung. Die wesentliche Eigenschaft einer Chipkarte ist, dass sie für Daten und Programme eine sichere Umgebung bietet.

1 Angriffe und Abwehrmaßnahmen während der Entwicklung

Schon in der Phase der Entwicklung von Mikrocontroller-Hardware und der Software für das Chipkarten-Betriebssystem werden die unterschiedlichsten Sicherheitsmaßnahmen ergriffen. Sicherheit muss, analog der Qualität, von Anfang an bei einer Entwicklung berücksichtigt werden und lässt sich nicht nachträglich in ein Produkt hineindesignen.

1.1 Entwicklung des Chipkarten-Mikrocontrollers

Die hardwaretechnische Entwicklung eines Chipkarten-Mikrocontrollers dauert viele Monate und wird von wenigen Personen bei einem Halbleiterhersteller in zutrittsgesicherten und überwachten Räumen durchgeführt. Die entsprechenden Computersysteme für das Halbleiterdesign sind in der Regel in ein autarkes und vom Rest der Welt abgekoppeltes Netzwerk eingebunden. Damit wird verhindert, dass von außen Veränderungen am Chipdesign vorgenommen oder der interne Aufbau des Chips ermittelt werden kann.

Um Manipulationen, die sich schwächend auf die Sicherheit auswirken, an einem Chipdesign vornehmen zu können, bedarf es sehr umfangreichen Insiderwissens, deshalb ist solch ein Angriff ziemlich unwahrscheinlich. Zudem werden mittlerweile beinahe alle Chipkartenhalbleiter von unabhängigen Prüfinstituten in Aufbau und Schutzmaßnahmen evaluiert.

Schutz: Designkriterien
Für die Gestaltung der Funktionen eines Chipkarten-Mikrocontrollers gibt es einige grundlegenden Kriterien. Zum einen müssen die Maßnahmen gegen statische und dynamische Angriffe auch wirksam sein. Sensoren und Schutzelemente nützen wenig, wenn sie allzu leicht umgehbar sind oder unter Umständen nie wirksam werden. Als Beispiel seien hier Sensoren auf dem Halbleiterchip aufgeführt, die einen so großen Flächenbedarf haben, dass sie ohne Probleme mit einer Nadel zerstört werden können und somit ihrer Schutzfunktion nicht mehr nachkommen.

Ein von den üblichen Standardbausteinen unterschiedliches, doch sehr wichtiges Designkriterium ist die Forderung, dass keinesfalls undokumentierte Mechanismen oder Funktionen (*„that's not a bug, that's a feature“*) vorhanden sein dürfen. Solche undokumentierten „Features“ werden meistens nie vollständig getestet, da sie nur wenige Personen kennen, und weisen insofern oft noch diverse Fehler und Schwachstellen auf. Da sie nicht dokumentiert sind, könnten sie versehentlich bei der Evaluierung der Hardware übersehen und später unter Umständen für Angriffe genutzt werden. Deshalb ist die Verwendung solcher undokumentierten Features streng untersagt, auch wenn sie für die Entwickler oftmals hilfreich sein könnten.

Schutz: Eindeutige Chipnummer
Bei der Entwicklung des Halbleiters müssen alle hardwaretechnischen Sicherheitselemente konzipiert und für den späteren Mikrocontroller umgesetzt werden. Neben den Sensoren und Schutzschichten ist ein Element dabei ein WORM-Speicher (*write once, read multiple*), oft auch OTP-Speicher (*one time programmable*) genannt. In diesen wird bei der Halbleiterproduktion eine eindeutige Chipnummer geschrieben. Damit ist der Chip individualisiert, auch eindeutig weiterverfolgbar, und die spätere Chipkarte kann dann systemweit eindeutig identifiziert werden. Zudem lässt sich diese Nummer auch zur Ableitung von Schlüsseln verwenden und bietet die Möglichkeit, Sperrlisten anzulegen, mit denen verdächtige Chipkarten aus dem Verkehr gezogen werden können.

Man darf dabei jedoch nicht übersehen, dass sich diese Nummer zwar nicht mehr auf einem Originalchip ändern lässt, doch schützt sie natürlich nicht vor einer Nachahmung des Chips durch einen frei programmierbaren Mikrocontroller. Deshalb dürfen Sicherheitsmechanismen nicht darauf beruhen, dass ein bestimmter Chip eine bestimmte Nummer in seinem WORM-Speicher hat. Diese eindeutige Nummer kann nur eine Grundlage für echte kryptografische Sicherheitsmechanismen sein.

1.2 Entwicklung des Chipkarten-Betriebssystems

Die Entwicklung von Software für Chipkarten verläuft analog den modernen Entwicklungsprinzipien von Software. Unabhängig davon, welche Entwicklungsmethodiken (Wasserfallmodell, Spiralmodell, …) verwendet werden, sind bestimmte Rahmenbedingungen einzuhalten.

Für die Entwicklungsrechner ist auf jeden Fall ein eigenes, vollständig abgeschottetes Netzwerk erforderlich, das keine Zugriffe von außen zulässt. Die Entwicklungswerkzeuge, wie Compiler und Chipsimulatoren, sind Software-Pakete, deren Funktionsfähigkeit in eigenen Tests geprüft wurde. Manchmal werden sogar zwei unterschiedliche Compiler benutzt, um sicherzustellen, dass das Ergebnis korrekt ist. Die Benutzung von Software, deren Herkunft nicht exakt nachweisbar ist, ist grundsätzlich untersagt, da dies ein möglicher Weg wäre, um ein Entwicklungswerkzeug zu manipulieren und dadurch in Folge den zu erstellenden Programmcode zu verändern.

Schutz: Entwicklungsprinzipien
Bei der Softwareentwicklung dürfen, ähnlich wie bei der Hardwareentwicklung, keine undokumentierten Features eingebaut werden. So wären beispielsweise, um die bei Chipkarten üblichen, aber aufwendigen Blackbox-Tests in Whitebox-Tests umzuwandeln, manchmal durchaus Kommandos angebracht, mit denen beliebige Speicherbereiche ausgelesen werden können. Würde jedoch eines dieser Kommandos im Programm vergessen, ließen sich damit bei echten Chipkarten geheime Schlüssel auslesen. Um diesen Angriff schon im Ansatz zu unterbinden, ist die Erstellung von Dump-Kommandos unerwünscht,

selbst wenn damit teure Entwicklungszeit gespart werden könnte. Termindruck und die stetig steigende Komplexität der Chipkarten-Betriebssysteme haben jedoch zur Aufweichung dieses Prinzips geführt. Um sicherzustellen, dass keinesfalls diese entwicklungsbegleitenden Kommandos in echten Chipkarten ins Feld kommen, werden in der Chipkarten-Fertigung spezielle Tests auf Nichtvorhandensein dieser Kommandos durchgeführt.

Ein weiteres Prinzip besteht darin, dass ein Programmierer niemals alleine an einer Programmentwicklung arbeitet. Dies verbietet sich schon aus Sicht der Software-Qualitätssicherung, doch auch aus Gründen der Sicherheit gegenüber Angriffen muss hier immer das Vier-Augen-Prinzip gelten. Damit lassen sich Angriffe durch Insider wirkungsvoll erschweren, da sich immer mindestens zwei Entwickler in ihrem Tun einig sein müssen. Zusätzlich werden noch regelmäßig interne Source-Code-Inspektionen durchgeführt, die zur Sicherstellung der Qualität und Überwachung des Entwicklungsprozesses dienen.

Ist die Softwareentwicklung abgeschlossen, findet häufig eine Prüfung des gesamten erstellten Sourcecodes und auch der Funktionalität durch unabhängige Prüfinstitute im Rahmen einer Software-Evaluierung statt. Die Gründe für diese zeit- und kostenintensiven Prüfungen liegen vor allem in der Kontrolle auf Softwarefehler, jedoch bewirken sie auch, dass es einem Entwickler nicht möglich ist, zum Beispiel ein trojanisches Pferd im Betriebssystem zu verstecken. Diese lassen sich in der Praxis eigentlich nur mittels Durchsicht des kompletten Programmcodes finden, da ein erfahrener Programmierer durchaus in der Lage wäre, Mittel und Wege zu finden, um ein trojanisches Pferd so zu verstecken, dass es durch Blackbox-Tests nicht mehr gefunden werden kann.

Schutz: Aufteilung von Wissen

Arbeiten mehrere Personen an einer Aufgabe, ist das Ergebnis aufgrund der unterschiedlichen Erfahrungen und Meinungen wesentlich robuster gegenüber Angriffen. Das Prinzip der Wissensteilung (*shared secret*) wirkt diesem Ansatz des „jeder weiß alles über alles“ entgegen. Grundsätzlich sollte bei der Entwicklung von Sicherheitskomponenten nicht das gesamte Wissen auf eine einzelne Person konzentriert sein, da sie damit auch angreifbar wird. So wird, ähnlich wie in manchen militärischen Bereichen, das Wissen bei der Entwicklung auf mehrere Gruppen von Personen verteilt, sodass zwar Diskussionen zwischen Experten über ein Thema möglich sind, doch es niemanden gibt, der alles weiß.

Analog zu Obigem verhält es sich mit der Komplettierung des Chipkarten-Betriebssystems, d. h. dem Laden von Tabellen, Programmcode und Konfigurationsdaten ins EEPROM. Dies stellt neben der erhöhten Flexibilität auch einen Sicherheitsaspekt dar. Das gesamte Wissen über das Betriebssystem befindet sich dann beim Chiphersteller, der den kompletten und assemblierten ROM-Programmcode zur Maskenherstellung erhält. Die Teile des Betriebssystems, die sich im EEPROM befinden, sind dem Chiphersteller nicht bekannt, sodass er bei einer Analyse des ROM-Codes nicht die gesamten Sicherheitsmechanismen und die Funktionalität des Betriebssystems erfahren kann.

2 Angriffe und Abwehrmaßnahmen während der Produktion

Angriffe während der Produktion der Chips bzw. der Chipkarten sind typische Insiderangriffe, da die betreffenden Produktionsumgebungen geschlossene Umgebungen sind. Der Zugang ist stark reglementiert, und jeder Zutritt wird aufgezeichnet. Trotzdem darf man die Produktion nicht bei den Sicherheitsbetrachtungen aussparen, da hier auch einige technisch sehr interessante und wirkungsvolle Angriffe durchgeführt werden könnten.

Schutz: Authentisierung bei den Fertigungsschritten

Schon bei der Waferherstellung sind die Chipkarten-Mikrocontroller durch eine Chipnummer individualisiert und mit einem Transportcode geschützt. Bei den neueren Be-

triebssystemen ist der Transportcode chipindividuell, und bei jedem fertigungstechnischen Zugriff auf den Chip ist eine Authentisierung zwingend erforderlich. Dies macht zwar die Fertigung etwas aufwendiger und erfordert natürlich ein Sicherheitsmodul an den jeweiligen Maschinen, doch erhöht es in erheblichem Maße die Sicherheit.

Ein nahe liegender Angriff in der Fertigung ist die Einschleusung von Dummy-Chips oder Dummy-Chipkarten, die sich identisch den regulären Bauteilen verhalten, jedoch beispielsweise ein Kommando zum Speicherdump haben. Der Austausch eines echten Chip durch einen Dummy-Chip ist natürlich frühestens nach der Trennung der Wafer in einzelne Dice möglich. Anhand einer Chipkarte für digitale Signaturen soll dieser Angriff hier kurz verdeutlicht werden: Der Angreifer tauscht bei der Initialisierung eine echte Chipkarte gegen eine Dummy-Karte aus. Diese Karte wird dann mit echten Daten initialisiert und anschließend personalisiert.

Da diese Chipkarte alle Funktionen einer echten Chipkarte aufweist, würde dann auch die Generierung der Schlüssel für den asymmetrischen Kryptoalgorithmus im Mikrocontroller durchgeführt. Die dafür notwendigen Daten bekommt er über die Initialisierungs- und Personalisierungsdaten. Anschließend müsste es der Angreifer noch schaffen, seine Chipkarte wieder in Besitz zu bekommen, und er könnte den geheimen Signaturschlüssel aus der Karte mit seinem speziellen Dumpkommando auslesen. Da der dazugehörige öffentliche Schlüssel vom Trustcenter unterschrieben wurde und damit als echt bestätigt ist, ist der Angreifer nun im Besitz aller Informationen, um beliebig viele Duplikate einer als echt anerkannten Karte herzustellen.

Solch ein Angriff ist schon deshalb unrealistisch, weil bereits in organisatorischer Hinsicht verhindert wird, Chips oder Chipkarten in die entsprechenden Fertigungsstätten hinein- oder herauszubringen. Zusätzlich verhindert die bei allen Fertigungsschritten notwendige Authentisierung zwischen Chipkarte und dem Sicherheitsmodul der Fertigungsmaschine einen Tausch der Chips oder Karten.

3 Angriffe und Abwehrmaßnahmen während der Kartenbenutzung

Der Zugang zur anzugreifenden Komponente – die Chipkarte – erfordert für Angreifer nach Ausgabe der Chipkarten in aller Regel sehr viel weniger Aufwand als in den vorangehenden Phasen des Lebenszyklus. Dies ist einer der Gründe, warum gerade in der Phase der Kartenbenutzung die Wahrscheinlichkeit von Angriffen relativ groß ist.

Im folgenden Abschnitt sind beispielhaft einige fast schon klassisch zu nennende Angriffe aufgeführt und erläutert. Die Angriffsbeschreibungen repräsentieren den Stand der Technik und sollen vor allem in der Thematik der Chipkartensicherheit Unerfahrenen einen kompetenten Überblick geben, damit bereits bekannte kritische Mechanismen nicht aus Unkenntnis abermals verwendet werden. Um sich gegen diese Angriffe zu verteidigen, gibt es die beschriebenen Abwehrmaßnahmen. Diese können wiederum mit etwas veränderten Angriffsszenarien umgangen werden, woraus sich das bekannte Katz- und Mausspiel von Maßnahmen und Gegenmaßnahmen bei Angriff und Abwehr ergibt.

Die dargestellten Szenarien sind keine Anleitung zum Brechen der Sicherheit eines Chipkarten-Systems, da sie ausnahmslos bekannt und auch öffentlich sind [Kömmerling 99]. Für die Sicherheit heutiger moderner Chipkarten stellen sie keine ernst zu nehmende Bedrohung dar, da die beschriebenen Angriffe durch entsprechende Schutzmaßnahmen längst berücksichtigt wurden. Vor einigen Jahren hätte man damit aber vielleicht noch einige Erfolge erzielt.

Tabelle 1 Überblick von typischen Angriffen, die auf Systeme mit Chipkarten Einfluss hatten, geordnet nach Zeitpunkt des Bekanntwerdens. Im folgenden Text sind die genannten Angriffe und die dazugehörigen ersten Gegenmaßnahmen detaillierter beschrieben.

bekannt seit	Angriff	Kurzbeschreibung des Angriffs
vor 1990	Abhören der Datenübertragung	Durch an das Modul angebrachte Drähte ist es möglich, die Datenübertragung zwischen Terminal und Karte abzuhören. Die Gegenmaßnahme dazu war die Einführung von Secure Messaging.
≈ 1990	Auflösen der Passivierung	Das Auflösen der Passivierungsschicht über dem Mikrocontroller ist die Voraussetzung für physischen Zugang zu den Komponenten auf dem Mikrocontroller. Die Gegenmaßnahme dazu war die Einführung von Passivierungsdetektoren auf den Mikrocontrollern.
≈ 1990	Manipulation der Datenübertragung	Durch elektrische Isolierung der Kontaktfelder des Moduls und entsprechend am Modul angebrachte Drähte ist es möglich, die Datenübertragung zwischen Terminal und Karte beliebig zu manipulieren. Die Gegenmaßnahme dazu war die Einführung von Secure Messaging.
≈ 1991	Löschen des EEPROM durch UV-Licht	Durch Löschen des EEPROMs mittels UV-Licht können beispielsweise Zähler wieder auf ihren Ausgangswert zurückgesetzt werden. Die Gegenmaßnahme dazu war die Einführung von Lichtsensoren auf den Mikrocontrollern.
≈ 1991	Ersatzschaltung von Speicherkarten	Mittels Ersatzschaltungen von Speicherkarten kann sowohl die Funktionalität der Speicherkarte als auch das geheime Echtheitsmerkmal emuliert werden. Die Gegenmaßnahme dazu war die Einführung von Challenge-Response-Authentisierung auf den Speicherkarten.
≈ 1992	Abschalten der Spannungsversorgung	Durch Abschalten der Spannungsversorgung bei der PIN-Prüfung kann das Schreiben des Fehlbedienungszählers verhindert werden. Die Gegenmaßnahme dazu war, den Fehlbedienungszähler vor der PIN-Prüfung prophylaktisch zu erhöhen.
≈ 1993	Takt anhalten	Durch Anhalten der Taktfrequenz und Analyse des RAM mittels Elektronenstrahltester können Rückschlüsse auf den RAM-Inhalt gezogen werden. Die Gegenmaßnahme dazu war die Einführung von Unterfrequenzdetektoren auf den Mikrocontrollern.
≈ 1993	Manipulation des Mikrocontrollers durch Lasercutter	Die Komponenten auf dem Mikrocontroller können mit Lasercutter manipuliert werden. Die Gegenmaßnahme dazu war die Einführung von Schutzschichten über den Mikrocontrollern.
1995	Timing-Attack	Aufgrund von Unwissen wurde bei der Implementierung vieler Kryptoalgorithmen eine Abhängigkeit zwischen Schlüssel und Laufzeit geschaffen. Dies kann zur Ermittlung der geheimen Schlüssel genutzt werden. Die Gegenmaßnahme dazu war die Realisierung von rauschfreien Kryptoalgorithmen.
≈ 1995	Abhören des Bus mit Mikroprobennadeln	Die Busse auf dem Mikrocontroller können mit Mikroprobennadeln abgehört werden. Die Gegenmaßnahme dazu war das Scrambling der Busse auf den Mikrocontrollern.
1996	DFA	Geheime Schlüssel von Kryptoalgorithmen können durch selektive Einstreuung von Fehlberechnungen des Prozessors berechnet werden. Die Gegenmaßnahmen dazu war die Einführung von Glitchdetektoren auf den Mikrocontrollern sowie entsprechende Vorsorgemaßnahmen in den Kryptoalgorithmen.
≈ 1996	Manipulation des Mikrocontrollers durch FIB	Der Komponenten auf dem Mikrocontroller können mit FIB manipuliert werden. Die Gegenmaßnahme dazu war die Einführung von Schutzschichten über dem Mikrocontroller.
1997	erschöpfende Schlüsselsuche beim DES	Durch leistungsfähige Rechner bzw. Netzwerke von Rechnern können DES-Schlüssel innerhalb einiger Stunden durch eine Brute-force-Attack errechnet werden. Die Gegenmaßnahme dazu war die Verwendung von Triple-DES.
1997	statistische Verteilung	Die Generierung der vierstelligen PINs im deutschen ec-

	von PIN	Kartensystem wies keine statistische Gleichverteilung auf, weshalb manche PIN-Werte deutlich öfter vorkamen als andere. Die Gegenmaßnahme dazu war die Verwendung eines verbesserten Generierungsalgorithmus.
1998	SPA/DPA	Aufgrund der Stromaufnahme des Prozessors lassen sich die verarbeiteten Daten ermitteln. Die Gegenmaßnahmen dazu waren: Einführung von zufallsgesteuerten Wartezeiten des Prozessors, Prozessoren mit immer gleicher Stromaufnahme sowie eine große Zahl von Vorsorgemaßnahmen in der Software auf dem Mikrocontroller.
1998	COMP 128	Aufgrund einer Designschwäche des von einigen Netzbetreibern verwendeten Authentisierungsalgorithmus COMP 128 ist es möglich, durch eine Brute-force-Attack die geheimen Schlüssel zu ermitteln. Die Gegenmaßnahme dazu war die Verwendung von anderen Authentisierungsalgorithmen und die Begrenzung der Anzahl der Authentisierungen.
1998	Störung des Prozessors	Durch Störung des Prozessors (z. B. durch Lichtblitze) durch beispielsweise Lichtblitze kann dieser bei der Abarbeitung des Maschinenprogramms an kritischen Stellen gestört werden. Die Gegenmaßnahmen dazu waren entsprechende Detektoren auf den Mikrocontrollern sowie eine große Zahl von Vorsorgemaßnahmen in der Software.

3.1 Angriffe auf der physikalischen Ebene

Für Manipulationen im Bereich des Halbleiters ist ein großer technischer Aufwand nötig. Je nach Angriffsszenario sind dies Mikroskope, Lasercutter, Mikromanipulatoren, fokussierte Ionenstrahlen, Anlagen für chemische Abtragverfahren und leistungsfähige Rechner für die Analyse, Protokollierung und Auswertung der elektrischen Vorgänge auf dem Chip. Diese Geräte und das Wissen um deren Anwendung stehen nur sehr wenigen Spezialisten oder Organisationen zur Verfügung, was die Wahrscheinlichkeit eines Angriffs auf physikalischer Ebene stark reduziert. Doch muss ein Karten- bzw. Halbleiterhersteller grundsätzlich davon ausgehen, dass ein potenzieller Angreifer alle notwendigen Geräte und Vorrichtungen für einen solchen Angriff einsetzen kann und dementsprechende Sicherungen in die Hardware einbauen.

Nachstehend sind die wichtigsten und in der Praxis am häufigsten angewandten Schutzmechanismen von Chipkarten-Mikrocontrollern erläutert.

3.1.1 Statische Analysen am Chipkarten-Mikrocontroller

Schutz: Halbleitertechnologie

Die Strukturen auf dem Chip (Breite der Leiterbahnen, Größe der Transistoren etc.) bewegen sich an der Grenze des derzeit technisch Machbaren. Die üblichen Strukturbreiten liegen im Bereich zwischen 0,35 µm und 0,13 µm, was an sich keine technische Besonderheit mehr darstellt. Die Transistordichte auf dem Silizium gehört jedoch zu den höchsten, die momentan mit den üblichen lithographischen Herstellungsverfahren möglich sind. Alleine diese sehr feinen Strukturen machen es schwierig, mit Analyseverfahren Informationen aus dem Chip herauszuholen, weshalb Halbleitertechnologien mit Strukturgrößen von einem Mikrometer und darunter zur Zeit noch als sicher angesehen werden. In Zukunft wird sich dies gewiss reduzieren.

Schutz: Chipdesign

Für das Design von halbleitertechnischen Bauelementen werden oft so genannte Standardzellen benutzt, die beispielsweise einen Prozessorkern oder bestimmte Speichertypen beinhalten. Der Vorteil besteht darin, dass ein Halbleiterhersteller mit diesen Standardelementen schnell und in hoher Qualität eine Vielfalt von unterschiedlichen Chips erstellen kann. Dieses für sicherheitsunkritische Massenartikel entwickelte Verfahren wird bei Chipkarten-Mikrocontrollern nicht eingesetzt, da Standardzellen in ihrem Aufbau und ihrer Funktionsweise bekannt sind und so einem potenziellen Angreifer zu viel Information in die Hände gegeben und somit die Arbeit wesentlich erleichtert würde.

Schutz: Busse auf dem Chip
Alle internen Busse auf dem Chip, die den Prozessor mit den drei verschiedenen Speichertypen ROM, EEPROM und RAM verbinden, sind nicht nach außen geführt und dadurch nicht kontaktierbar. Es besteht für einen Angreifer keine Möglichkeit, den Adress-, Daten- oder Steuerbus des Mikrocontrollers abzuhören oder zu beeinflussen und dadurch Speicherinhalte auszulesen. Überlicherweise werden die Busse in den unteren Schichten des Halbleiters aufgebaut, sodass es schwierig ist, sie direkt von der Oberfläche her zu kontaktieren. Zudem sind die Busse auf dem Chip statisch, chipindividuell oder sitzungsindividuell gescrambelt, sodass die Funktion der einzelnen Busleitungen von außen nicht erkennbar ist. Es gibt Chipkarten-Mikrocontroller bei denen das Scrambling der Busse sogar während der Sitzung laufend geändert wird.

Schutz: Speicherdesign
Das Speichermedium für die meisten Programme ist das ROM. Der Inhalt eines in der Industrie im Allgemeinen verwendeten ROMs kann mit einem Lichtmikroskop Bit für Bit gelesen werden. Diese Bits zu Bytes zusammenzusetzen und die Bytes zu dem kompletten ROM-Code, stellt dann keine große Schwierigkeit mehr dar. Um genau diese Analyse zu unterbinden, befindet sich das ROM nicht in den obersten und somit am leichtesten zugänglichen Schichten, sondern in den unteren Siliziumschichten. Damit wird eine optische Analyse verhindert.

Würde man aber den Chip mit seiner Vorderseite auf einen Träger aufkleben und dann von seiner Rückseite her abschleifen, könnte man wiederum die ROM-Inhalte auslesen. Um dem vorzubeugen benutzt man bei den Chipkarten-Mikrocontrollern nur ionenimplantiertes ROM, dessen Dateninhalte weder im visuellen noch im IR- oder UV-Spektrum sichtbar sind. Dies schützt auch weitgehend vor so genanntem selektivem Ätzen. Bei diesem Verfahren wird versucht, den Halbleiter so zu ätzen, dass ROM-Inhalte wieder optisch sichtbar werden.

Schutz: Schutzschichten (*shield*)
Eine Gefahr ist die Analyse von elektrischen Potenzialen auf dem Chip während des Betriebes. Damit besteht bei genügend hoher Abtastfrequenz die Möglichkeit, Ladungspotenziale, d. h. Spannungen, auf sehr kleinen Kristallbereichen zu messen und so Rückschlüsse auf Dateninhalte des RAM während des Betriebes zu ziehen. Dies kann sehr zuverlässig durch Strom führende Metallisierungsebenen über den entsprechenden Speicherbereich oder den ganzen Chip verhindert werden. Entfernt man diese Metallschichten auf chemischem Weg, ist der Chip nicht mehr funktionsfähig, da er diese Schichten als elektrische Spannungszuführungen zur Funktion benötigt. Oft werden sogar mehrere Schutzschichten übereinander angeordnet und permanent auf Unversehrtheit geprüft.

Zusätzlich können halbleitertechnisch über die gesamte Chipoberfläche oder über besonders gefährdete Bereiche (z. B.: Unterfrequenzdetektor) mäanderförmige und Strom führende Strukturen gelegt werden. Sie lassen sich problemlos über Widerstands- oder Kapazitätsmessung messtechnisch überwachen oder in die Chipfunktion einbinden, sodass eine Beschädigung zum sofortigen Abschalten des Chips führt. Die Sicherheit kann dabei noch erhöht werden, indem die Verschaltung der mäanderförmigen Strukturen während einer Sitzung geändert wird. Dies verhindert, dass die Mäander mittels FIB (*Focused Ion Beam*) überbrückt werden können.

Schutz: Scrambling des Speichers
Analog zu dem seit langem üblichen Scrambling von Bussen findet zunehmend auch ein Scrambling der Speicher auf den Mikrocontroller-Chips statt. Die Sicherheit beruht auf der Geheimhaltung des Verwürfelungsschemas der Speicherzellen. Das Speicherscrambling ist einfach zu realisieren und benötigt wenig zusätzlichen Platz auf dem Chip. Ohne die entsprechenden Scrambling-Informationen ist es für einen Angreifer extrem schwierig, herauszufinden, wie die Speicherzellen durchadressiert sind.

Schutz: Verschlüsselung des Speichers
Neben dem Vertauschen von Daten im Speicher (*scrambling*) bieten moderne Chipkarten-Mikrocontroller auch eine chargen- oder chipindividuelle Verschlüsselung des Speichers und zum Teil auch der Register des Prozessors an. Dabei werden beim Lesen und Schreiben die entsprechenden Daten in Echtzeit ent- oder verschlüsselt. Zusätzlich zum Schlüssel kann bei manchen Chiptypen noch die Speicheradresse in den Ver-/Entschlüsselungsprozess einfließen, sodass gleiche Daten an unterschiedlichen Stellen im Speicher nach der Verschlüsselung unterschiedliche Werte haben. Speziell für RAM-Bereiche können auch sitzungsindividuelle Schlüssel benutzt werden.

Wenn durch einen erfolgreichen Angriff Daten aus dem Speicher gelesen werden könnten, würde man dazu noch zusätzlich den geheimen Schlüssel benötigen, um daraus den Klartext zu gewinnen. Dies erhöht den Aufwand des Angreifers erheblich, da er entweder wissen müsste, an welcher Stelle dieser Schlüssel gespeichert ist, oder pauschal alle auf dem Chip vorhanden Daten lesen müsste.

3.1.2 Dynamische Analysen am Chipkarten-Mikrocontroller

Schutz: Passivierungsüberwachung
Nach der Herstellung des Mikrocontrollers auf dem Silizium wird eine Passivierungsschicht aufgetragen, die Oxidation, beispielsweise durch Luftsauerstoff, oder weitere chemische Prozesse auf der Chipoberfläche verhindert. Um Manipulationen auf dem Chip vorzunehmen, muss immer als Erstes diese Passivierungsschicht entfernt werden. Es sollte jedoch bedacht werden, dass sich die Passivierungsschicht zwar chemisch abtragen lässt, er Chip aber dann einer großen Oxidationsgefahr unterliegt, die ihn relativ schnell zerstören kann. Eine Sensorschaltung kann über Widerstands- oder Kapazitätsmessung feststellen, ob diese Passivierungsschicht noch vorhanden ist. Ist sie nicht mehr existent oder beschädigt, kann entweder ein Interrupt in der Chipsoftware ausgelöst werden, oder der gesamte Chip wird von der Hardware abgeschaltet, was alle dynamischen Analysen zuverlässig verhindert.

Schutz: Spannungsüberwachung
Auf jedem Chipkarten-Mikrocontroller ist eine Spannungsüberwachung vorhanden. Diese sorgt für ein definiertes Abschalten des Bausteins, wenn die oberen oder unteren Grenzen der Betriebsspannung über- bzw. unterschritten werden. Auf diese Weise erhält die Software die Sicherheit, dass ein Betrieb in den Grenzbereichen, in denen der Chip nicht mehr voll funktionsfähig ist, unmöglich ist. Wäre keine Spannungsüberwachung vorhanden, kann es in diesen Grenzbereichen vorkommen, dass z. B. der Programmzähler des Prozessors nicht mehr stabil läuft, was beispielsweise zu unkontrollierten Sprüngen innerhalb des Programms führt oder schlichtweg zu Rechenfehlern im Prozessor. Dieses Fehlverhalten lässt sich als Ansatzpunkt zur Ermittlung von geheimen Schlüsseln mittels der weiter unten im Text beschriebenen differenziellen Fehleranalyse (DFA) nutzen.

Gerade die Spannungsüberwachung ist von sehr großer Bedeutung für die Sicherheit des Mikrocontrollers. Man könnte sich hier als Angriff vorstellen, dass in einem ersten Schritt die

entsprechenden Detektoren z. B. mittels FIB (*Focused Ion Beam*) unbrauchbar gemacht werden und dann in einem zweiten Schritt der eigentliche Angriff startet. Aus diesem Grund sind oft die für die Sicherheit eines Mikrocontrollers wichtigen Komponenten besonders geschützt, sodass eine Manipulation erkannt wird und sich die Chipkarte automatisch deaktiviert.

Schutz: Frequenzüberwachung
Die Taktversorgung der Chipkarte läuft in der Regel extern, sodass die interne Rechengeschwindigkeit von außen bestimmt wird. Damit besteht zumindest theoretisch die Möglichkeit, den Mikrocontroller im Einzelschrittbetrieb zu fahren. Dies würde zu hervorragenden Analysemöglichkeiten vor allem in der Messung von Stromaufnahme (*Power Analysis*) und elektrischen Potenzialen auf dem Chip führen. Um diesen Angriff zu verhindern, ist auf dem Chip eine Funktionsbaugruppe zur Unter- und Überfrequenzdetektion integriert. Diese unterbindet, dass der angelegte Takt in unzulässiger Weise erniedrigt werden kann.

Zum Schutz vor dem gefährlichen Einzelschrittbetrieb des Mikrocontrollers ist es sinnvoll, den Unterfrequenzdetektor durch Schutzschichten abzusichern, sodass sich dieser keinesfalls unbemerkt manipulieren lässt.

Schutz: Scrambling der Busse
Viele Chipkarten-Mikrocontroller scrambeln die nur intern auf dem Chip zugänglichen Busse zur Ansteuerung der Speicher. Dies bedeutet, dass die einzelnen Busleitungen nicht in auf- oder absteigender Folge, sondern wirr und mehrfach gegeneinander vertauscht nebeneinander oder sogar getrennt durch Isolationsschichten übereinander angeordnet sind. Für den potenziellen Angreifer ist dies eine zusätzliche Hürde, da er nicht weiß, welche Busleiterbahn welche Funktion bzw. Adresse hat.

Dieses Mischen der Leiterbahnen wurde ursprünglich nur in einer statischen Variante eingeführt, d. h. auf jedem Chip sind die Vertauschungen identisch. Damit wäre es wahrscheinlich mittelfristig für einen Angreifer kein größeres Problem, die Vertauschung herauszufinden und bei einer Abhöraktion entsprechend zu berücksichtigen.

Es gibt hier aber eine Verbesserung der Sicherheit, indem man ein chipindividuelles Scrambling der Busse einführt. Dieses chipindividuelle Scrambling kommt nicht zustande, indem man für jeden Chip eigene Belichtungsmasken für die Busse erstellt, da dies technisch momentan nicht realisierbar bzw. unbezahlbar wäre. Das Scrambling wird dabei durch direkt an den Speichern sitzende Verwürfler vorgenommen, die beispielsweise über die chipindividuelle Nummer angesteuert werden. Dies ist halbleitertechnisch ohne großen Aufwand möglich und erschwert das Abhören beträchtlich. Ein chipindividuelles und sitzungsspezifisches Scrambling ist durch variable Eingangswerte der Verwürfler somit ebenfalls realisierbar.

Dynamische Analyse und Abwehr: Messen des Stromverbrauchs der CPU
Im Juni 1998 veröffentlichten Paul Kocher, Joshua Jaffe und Benjamin Jun ein Dokument über einfache Leistungsanalyse (*simple power analysis – SPA* und differenzielle Leistungsanalyse (*differenzial power analysis – DPA*) veröffentlichten [Kocher 98a].

Das Prinzip der einfachen Leistungsanalyse (SPA) ist relativ einfach. Mit einem Analog-Digital-Wandler wird anhand des Spannungsabfalls an einem seriell vorgeschalteten Widerstand mit hoher zeitlicher Auflösung der Stromverbrauch eines Mikrocontrollers gemessen. Der relativ einfache Aufbau der CPUs von Chipkarten-Mikrocontrollern führt dazu, dass die internen Abläufe und verarbeiteten Daten zu messbaren und auch interpretierbaren Auswirkungen auf den Stromverbrauch führen. Zur Verdeutlichung kann man sich vorstellen, dass der gleiche Programmablauf mit gleichen Daten zu einem bestimmten zeitlichen Verlauf des Stromverbrauchs des Prozessors führt. Wird dieses Programm nun mit anderen Daten durchlaufen, unterscheidet sich der zeitliche Verlauf des Stromverbrauchs. Diese Abweichung verwendet man nun zur Ermittlung der in dem Ablauf prozessierten Daten.

Mit der differenziellen Leistungsanalyse (DPA) können gegenüber der einfachen Leistungsanalyse (SPA) noch geringere Unterschiede beim Stromkonsum des Mikrocontrollers aufgedeckt werden. Dazu misst man den Strom zuerst bei der Abarbeitung bekannter und

anschließend unbekannter Daten. Die Messungen werden in der Regel viele Male wiederholt, damit durch Mittelwertbildung das Rauschen eliminiert werden kann. Im Anschluss an diese Messungen bildet man die Differenz und kann von diesem Ergebnis auf die unbekannten Daten schließen.

Die Leistungsanalyse für Chipkarten-Mikrocontroller sind für unvorbereitete Hard- und Software sehr ernst zu nehmende Angriffe. Der Grund dafür besteht darin, dass bei manchen Mikrocontrollern durchaus Abhängigkeiten des Stromverbrauchs vom jeweiligen Maschinenbefehl und auch von den in diesem Maschinenbefehl verarbeiteten Daten bestehen. Zudem hält sich der für einem erfolgreichen Angriff erforderliche Aufwand an Messgeräten in Grenzen. Es existiert aber eine Reihe wirkungsvoller Gegenmaßnahmen, die sich einerseits auf entsprechend verbesserte Hardware und andererseits auf modifizierte Software stützen.

Die einfachste Hardware-technische Lösung ist der Einbau eines schnellen Spannungsreglers auf dem Chip, der über einen Shunt-Widerstand einen von Maschinenbefehl und Daten unabhängigen Stromverbrauch sicherstellt. Künstliche Stromrauschquellen auf dem Chip sind ebenfalls eine wirksame Lösung. Eine technisch aufwendigere Lösung besteht in einem modifizierten halbleitertechnischen Design des Prozessors, damit dieser immer einen konstanten Stromverbrauch aufweist. Diese Lösungsansätze erhöhen jedoch zum Teil den Stromverbrauch des Mikrocontrollers, was in bestimmten Anwendungsfeldern wie der Telekommunikation nicht gewünscht ist. Eine einfache Abwehrmaßnahme kann auch darin bestehen, während eines SPA/DPA-kritischen Vorgangs für diesen Vorgang nicht benötigte Komponenten wie CRC-Prüfsummengenerator oder den numerischen Coprozessor mit Zufallsdaten als Eingangswerte zu aktivieren, um auf diese Weise ein künstliches Rauschen des Stromverbrauchs zu erzeugen.

Die Verwendung von zufallsgesteuerten Wartezeiten (*random wait states*) beim Prozessor erschwert die Synchronisation bei der Stromanalyse erheblich, ohne dabei die Nach-teile eines erhöhten Stromverbrauchs in Kauf zu nehmen. Ein ähnlicher Lösungsansatz wird bei Chipkarten-Mikrocontrollern mit eigener Takterzeugung auf dem Chip verfolgt, indem die Taktfrequenz zufallsgesteuert innerhalb vorgegebener Grenzen permanent variiert wird.

Bei den softwaretechnischen Gegenmaßnahmen gibt es mittlerweile eine immense Bandbreite an Lösungsvarianten. Im Folgenden seien einige wenige davon stellvertretend kurz aufgeführt. Der einfachste Ansatz ist die ausschließliche Benutzung von Maschinenbefehlen mit einem sehr ähnlichen Stromverbrauch. Maschinenbefehle, welche eine signifikante Abweichung vom durchschnittlichen Strombedarf haben, dürfen dann nicht mehr im Assemblercode verwendet werden. Eine weitere Lösungsmöglichkeit besteht darin, für gleiche Berechnungen in Kryptoalgorithmen verschiedene Abläufe einzuführen, welche jeweils zufällig ausgewählt werden. Dies erschwert es dem Beobachter erheblich, eine Konvergenz zwischen bekannten und unbekannten Maschinenbefehlen sowie verarbeiteten Daten zu erkennen. Eine Abwehr in ähnlicher Richtung wird mit chipindividuellen Tabellen für die S-Boxen des Triple-DES-Algorithmus verfolgt.

Um die zu einer erfolgreichen Stromanalyse vorab benötigte Datenerfassung zu erschweren, sollten alle Schlüssel mit irreversibel ablaufenden Fehlbedienungszählern abgesichert sein. Weiterhin ist es notwendig, den freien Zugriff auf alle Kommandos in der Art von INTERNAL AUTHENTICATE, bei denen beliebige Daten durch einen Kryptoalgorithmus der Chipkarte geschickt werden können, zu sperren. Durch diese Einschränkung der Kommandobenutzung verhindert man ebenfalls die Sammlung von Referenzdaten für die spätere Stromanalyse.

Analyse und Abwehr:
Messen der elektromagnetischen Abstrahlung der CPU

Zumindest theoretisch könnte man analog zur differenziellen Leistungsanalyse anhand der elektromagnetischen Abstrahlung Schlüsse auf interne Abläufe auf dem Chipkarten-Mikrocontroller ziehen. Mit SQUIDs (*superconducting quantum interference devices*) lassen sich Magnetfelder von geringer Ausdehnung und Stärke messen. Die Auswertung kann dann analog SPA/DPA erfolgen. Ein ersten Ansatz zu diesem Angriff wird bei Karine Gandolfi [Gandolfi 01] beschrieben. Allerdings ist der nötige technische Aufwand hoch und das erforderliche Wissen über die internen Strukturen des Halbleiters nicht allgemein verfügbar. Zudem können Halbleiterbausteine gegen diese Art von Angriffen sehr effektiv geschützt werden, indem man mehrere Leiterbahnen übereinander legt, sodass zwar ein Magnetfeld mit empfindlichen Detektoren gemessen werden kann, aber nicht, welche der übereinander liegenden Leitungen Strom führend ist.

3.2 Angriffe auf der logischen Ebene

Angriffe auf die Sicherheit einer Chipkarte auf logischer Ebene setzen vor allem ein Verständnis der Kommunikation und des Informationsflusses zwischen Terminal und Chipkarte voraus. Es ist nicht so sehr notwendig, Prozesse auf der Hardware-Ebene zu verstehen, wie die Abläufe in der Software. Die hier beispielhaft vorgestellten Szenarien bewegen sich informationstechnisch gesehen eine Ebene über den Angriffen, bei denen vor allem die Hardwareeigenschaften ausgenutzt werden.

Angriff und Abwehr: Dummy-Chipkarte
Der wohl am ehesten vorstellbare Angriff ist die Verwendung einer selbst programmierten und mit diversen Analyse- und Protokollfunktionen erweiterten Chipkarte. Bis vor einigen Jahren war dies so gut wie undurchführbar, da der Erwerb von Chipkarten bzw. den dazugehörigen Mikrocontrollern nur einigen wenigen Firmen möglich war. Doch mittlerweile sind Chipkarten und Konfigurationsprogramme von verschiedenen Firmen frei zu kaufen. Damit erweitern sich natürlich auch die Möglichkeiten, die einem Angreifer zur Verfügung stehen. Doch unabhängig davon lässt sich mit etwas Aufwand aus einem Kunststoffplättchen, einem Standard-Mikrocontroller im SMD-Gehäuse und etwas Geschick eine funktionsfähige Chipkarte zusammenbauen. Zumindest eine, die sich elektrisch und während der Datenübertragung wie eine echte verhält. Mittlerweile kann diese Art von Chipkarten über die unterschiedlichsten Händler via Internet erworben werden. Neue Möglichkeiten bietet auch die Java-Technologie bei Chipkarten, bei der problemlos eigene Programme erstellt und in eine Dummy-Karte geladen werden können.

Mit einer solchen Dummy-Karte ließe sich ein Teil der Kommunikation mitdem Terminal protokollieren und später auswerten. Nach mehreren Versuchen ist es dann wahrscheinlich möglich, einen Teil der Kommunikation wie mit einer echten Chipkarte auszuführen.

Ob man daraus auch einen Vorteil erzielen kann, ist zweifelhaft, da alle professionell gestalteten Anwendungen über eine kryptografische Absicherung für wichtige Aktionen verfügen. Solange man den geheimen Schlüssel nicht kennt, ist spätestens bei der Authentisierung das Ende des Angriffs erreicht. Dieser Angriff würde nur zum Erfolg führen, wenn einem der geheime Schlüssel bekannt ist oder die gesamte Anwendung ohne kryptografische Absicherung abläuft. Sollte eine solche Anwendung existieren, darf aber stark bezweifelt werden, ob der durch diesen Angriff erreichbare Vorteil von einer solchen Bedeutung ist, dass der ganze dafür notwendige Aufwand gerechtfertigt ist.

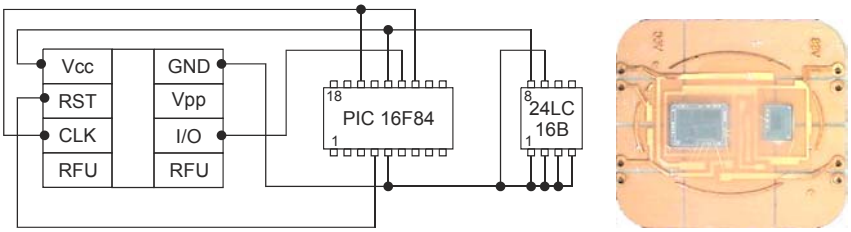

Bild 2 Typische Ersatzschaltung für einen Chipkarten-Mikrocontroller, aufgebaut aus diskreten Standardbauelementen (Mikrocontroller PIC 16F84, EEPROM-Speicherchip 24LC16B). Die Bauelemente passen in ein typisches Chipkartenmodul, sodass ohne Untersuchung des Moduls keine Unterscheidung von einem echten Chipkarten-Mikrocontroller möglich ist. Diese Schaltung und Varianten davon finden sich an den einschlägigen Stellen im Internet.

Angriff: Abhören der Datenübertragung

Um die Daten während einer Sitzung abzuhören und bei Bedarf manipulieren zu können, verwendet man eine leicht abgeänderte Chipkarte. Auf die I/O-Kontaktfläche wird dazu ein elektrisch isolierter Dummy-Kontakt aufgeklebt. Die ursprüngliche I/O-Schnittstelle ist mit ihm dann elektrisch nicht mehr verbunden. Der auf diese Weise neu geschaffene (Dummy-)Kontakt und der ursprüngliche I/O-Kontakt sind mit einem schnellem Computer verbunden. Dieser kann nun je nach Programmierung bei der Kommunikation zwischen Terminal und Chipkarte beliebige Daten ausschneiden oder einblenden. Ist der Computer hinreichend schnell, wird weder Terminal noch Chipkarte bei der manipulierten Kommunikation einen Unterschied zum regulären Datenaustausch feststellen können.

Man versteht, dass mit dieser Methode der Ablauf einer Sitzung massiv beeinflusst werden kann. Ob nun einem Angreifer daraus ein Vorteil erwächst, hängt vor allem von der Anwendung in der Chipkarte ab. Ein anerkanntes Designkriterium besagt, dass weder durch Abhören, Ausschneiden oder Einblenden von Daten während der Kommunikation die Sicherheit beeinträchtigt sein darf. Wird dieses Kriterium nicht beachtet, kann ein Angreifer auf diese Weise sicherlich einen Vorteil erlangen

Angriff und Abwehr: Abschalten der Stromversorgung

Ein Angriff, der noch vor einigen Jahren bei vielen Chipkarten zum Erfolg geführt hat, ist das Abschalten der Stromversorgung zu einem bestimmten Zeitpunkt während der Ausführung eines Kommandos. Der Hintergrund für diesen Angriff ist die Tatsache, dass bei konventioneller Programmierung alle Schreiboperationen auf EEPROM-Seiten nacheinander ausgeführt werden. Hat nun der Programmierer des Kommandos die Reihenfolge der Schreiboperationen unklug angeordnet, kann das Abschalten der Stromversorgung zum richtigen Zeitpunkt einem Angreifer zum Vorteil gereichen.

Die Betriebssystem-Designer kennen jedoch auch für diesen Angriff eine wirkungsvolle Gegenmaßnahme: atomare Abläufe. Sie besitzen die Eigenschaft, dass sie atomar, also nicht aufteilbar sind. Das bedeutet, sie werden entweder ganz oder gar nicht ausgeführt, was als Schutz für den obigen Angriff vollkommen ausreichend ist.

Angriff und Abwehr: Stromanalyse bei PIN-Vergleichen

Mit der Kombination von physikalischer Messung eines Parameters und Variation von logischen Werten kann ein technisch sehr interessanter Angriff auf Vergleichsmerkmale, wie z. B. die PIN, unternommen werden. Er betrifft alle Mechanismen, bei denen Daten zur Chipkarte gesendet und dort mit einem gespeicherten Wert verglichen werden, wobei abhängig vom Vergleichsergebnis ein Fehlbedienungszähler erhöht wird.

Das Prinzip ist dabei eine Strommessung der Chipkarte, die beispielsweise über den Spannungsabfall an einem in die Vcc-Leitung eingebrachten Widerstand erfolgt. Sendet

man das betreffende Kommando mit den Vergleichsdaten zur Karte, so kann man über die Strommessung vor Erhalt des Returncodes feststellen, ob der Fehlbedienungszähler erhöht wurde oder nicht. Würde nun bei einem positiven Vergleich der Returncode früher ausgesandt, als der Fehlbedienungszähler geschrieben wird, so könnte man auf dieser Grundlage den Vergleichswert ermitteln. Dazu sendet man den Vergleichswert in allen seinen Varianten zur Chipkarte und schaltet diese im Schlechtfall immer vor dem Erhöhen des Fehlbedienungszählers ab. Der Gutfall kann durch den entsprechenden Returncode, der vor der Erhöhung des Fehlbedienungszählers gesendet wird, eindeutig erkannt werden.

Um nun diesen Angriff abzuwehren, gibt es zwei grundsätzliche Methoden: Die einfachste Abwehr besteht darin, dass man den Fehlbedienungszähler grundsätzlich vor jedem Vergleich erhöht und ihn dann bei Bedarf wieder erniedrigt. Egal zu welchem Zeitpunkt nun ein Angreifer die Spannungsversorgung unterbricht, er kann daraus nie einen Vorteil erzielen, da der Fehlbedienungszähler schon erhöht ist. Die zweite Variante ist etwas aufwendiger, erfüllt aber die gleiche Schutzfunktion. Nach dem Vergleich wird im Schlechtfall der Fehlbedienungszähler erhöht und im Gutfall in eine nicht benutzte EEPROM-Zelle geschrieben. Beide Schreibzugriffe finden zeitgleich statt, sodass ein Angreifer keine Rückschlüsse auf den Vergleich ziehen kann. Er erfährt erst durch den Returncode vom Ergebnis der Vergleichsoperation. Zu diesem Zeitpunkt ist es dann schon zu spät, durch Abschalten der Spannungsversorgung den Schreibzugriff auf den Fehlbedienungszähler zu verhindern.

Angriff und Abwehr: Zeitanalyse bei PIN-Vergleichen

Programmierer achten immer darauf, dass Programme so schnell wie möglich ausgeführt werden. Im Regelfall ist dies auch wichtig. Allerdings lässt sich diese Tatsache der Laufzeitminimierung auch für einen durchaus erfolgversprechenden Angriff nutzen. Wird einer Chipkarte eine PIN zur Prüfung übergeben, so führt die zuständige Vergleichsroutine einen byteweisen Vergleich der übergebenen PIN und der abgespeicherten PIN aus. Ein nicht auf Sicherheit achtender Programmierer wird nun diese Vergleichsroutine so programmieren, dass ein Unterschied beim Vergleich der beiden PIN-Ziffern zum sofortigen Abbruch und Aussprung aus der Vergleichsroutine führt. Damit ergeben sich geringe, doch einem mit geeigneten Instrumentarium (z. B. Speicheroszilloskop) durchaus messbare Laufzeitunterschiede aufgrund des abgebrochenen Vergleichs. Diese können von einem Angreifer benutzt werden, um auf relativ einfache Art und Weise die normalerweise geheime PIN zu ermitteln.

Obiger Angriff war vor einigen Jahren im Bereich der Chipkarten noch erfolgreich. Mittlerweile ist aber diese Art von Angriffen bekannt, und die Vergleichsroutinen sind so ausgelegt, dass prinzipiell immer alle Stellen einer PIN verglichen werden. Damit tritt kein Zeitunterschied zwischen positivem und negativem Vergleichsergebnis auf.

Schutz: Rauschfreier Kryptoalgorithmus

Noch in den frühen 90er-Jahren wurden manchmal Kryptoalgorithmen verwendet, die erhebliche Unterschiede in der Ausführungszeit abhängig vom Schlüssel und Klartext hatten. Mit dem so reduzierten Schlüsselraum als Grundlage kann der Angreifer mit einem Bruteforce-Angriff nach dem geheimen Schlüssel suchen. Wie lange die Suche dauert, hängt sehr stark vom Rauschen des Algorithmus ab. Je größer aber die Zeitunterschiede sind, desto kleiner ist der Schlüsselraum und desto einfacher und schneller die Schlüsselsuche. Falls die exakte Implementation des entsprechenden Kryptoalgorithmus auf dem Zielrechner bekannt ist, kann dies zusätzlich als Referenz zur Erstellung von Zeittabellen herangezogen werden. Publik gemacht wurde diese Art des Angriffs – eine „Timing Attack“ – 1995 in einer Veröffentlichung von Paul Kocher [Kocher 95], die sich vor allem mit Zeitabhängigkeiten bei RSA und DSS beschäftigte.

Im Prinzip ist eine Timing Analysis für die Sicherheit einer Chipkarte sehr gefährlich. Da er aber schon seit längerer Zeit bekannt ist, benutzen alle heutigen Chipkarten nur noch

rauschfreie Kryptoalgorithmen, d. h. die Zeit für Ver- und Entschlüsselung ist unabhängig von den Eingangswerten. Damit wurde diese Art des Angriffs abgeblockt.

Als zusätzliche Sicherheit verfügen in manchen Anwendungen alle Authentisierungsschlüssel noch über eigene Fehlbedienungszähler, sodass sich nur eine bestimmte Anzahl nicht erfolgreicher Authentisierungen durchführen lässt. Hat der Fehlbedienungszähler seinen Maximalwert erreicht, sperrt sich die Chipkarte gegen alle weiteren Authentisierungsversuche.

Manipulation: Differenzielle Fehleranalyse (*differenzial fault analysis – DFA)*

Im Jahr 1996 veröffentlichten Dan Boneh, Richard DeMillo und Richard Lipton eine Arbeit [Boneh 96], die ein theoretisches Modell beschreibt, wie geheime Schlüssel von asymmetrischen Kryptoalgorithmen durch Einstreuung von Hardwarefehlern berechnet werden können.

Nur zwei Monate später publizierten Eli Biham und Adi Shamir eine Erweiterung des Bellcore-Angriffs mit dem Namen differenzielle Fehleranalyse (*differenzial fault analysis–DFA*) [Biham 96], die nunmehr auch symmetrische Kryptoalgorithmen, wie den DES-Algorithmus, mit einschloss. Damit waren, zumindest in der Theorie, viele Chipkarten-Anwendungen von einer neuen, ernst zu nehmenden Angriffsmethode betroffen.

Das grundlegende Prinzip beider Angriffe ist verhältnismäßig einfach: Im ersten Schritt verschlüsselt man einen beliebigen Klartext mit dem zu brechenden Schlüssel und bewahrt den erhaltenen Schlüsseltext auf. Anschließend wird die Chipkarte während der Abarbeitung des kryptografischen Algorithmus in ihrer Arbeitsweise von außen beispielsweise durch ionisierende oder hochfrequente Strahlung gestört, sodass sich ein einzelnes Schlüsselbit an beliebiger Stelle bei der Berechnung verändert. Das Ergebnis davon ist ein Schlüsseltext, der aufgrund des gekippten Bits falsch verschlüsselt wurde. Dies wird nun mehrmals wiederholt, und die Ergebnisse werden zur Analyse aufbewahrt. Der Rest zur Ermittlung des Schlüssels ist pure Mathematik und in den oben aufgeführten Arbeiten umfassend dargestellt.

Die Stärke des Angriffs liegt vor allem darin, dass es nicht einmal erforderlich ist zu wissen, an welcher Stelle des geheimen Schlüssels ein Bit gekippt wurde. Biham und Shamir geben in ihrer Veröffentlichung an, dass bei einem verfälschten Schlüsselbit bereits 200 Schlüsseltextblöcke genügen, um daraus den geheimen DES-Schlüssel zu berechnen. Bei Verwendung eines echten Triple-DES (168 Bit) anstelle des DES erhöht sich die Zahl der notwendigen Schlüsseltexte nur unwesentlich. Selbst wenn mehr als ein Bit verändert wird, greift dieser Angriff noch. Es erhöht sich lediglich die notwendige Anzahl der falsch verschlüsselten Schlüsseltexte.

So anspruchslos, wie sich diese Art von Angriffen anhört, ist sie in der Praxis dann doch nicht. Es sollten möglichst nur ein Bit oder zumindest sehr wenige Bits verändert werden. Sendet man aber nun pauschal auf den gesamten Mikrocontroller hochfrequente Mikrowellenstrahlung, ändern sich meist so viele Bits, dass der Prozessor in der Regel sofort rettungslos abstürzt. Deshalb versucht man beispielsweise durch vorsätzlich erzeugte Glitches[1] in der Strom- oder Taktversorgung, die CPU zu einer einzigen falschen Berechnung zu veranlassen. Wenn die Filter an den dazugehörigen Eingangsleitungen solch ein Glitch nicht neutralisieren können, kann es zu der beabsichtigten Fehlrechnung des Prozessors kommen.

Eine Chipkarte ist allerdings weder dem Bellcore-Angriff noch einer DFA schutzlos ausgeliefert, wenn zuvor entsprechende Vorsorge praktiziert wurde. Die einfachste Abwehr ist, den Kryptoalgorithmus in der Chipkarte einfach zweimal zu berechnen und die beiden Ergebnisse zu vergleichen. Sind die Resultate identisch, dann wurde nicht versucht, von außen irgendwelche Bits zu kippen. Dabei geht man davon aus, dass eine bewusste Fehlereinstreu-

[1] Glitches sind sehr kurze Spannungseinbrüche oder -erhöhungen

ung niemals die gleichen Bits in der Chipkarte verändern kann. Dies ist wirklichkeitsnah, denn sollte jemals die gezielte Änderung von bestimmten Bits in einem Chipkarten-Prozessor möglich sein, gibt es viel einfachere und schneller durchzuführende Angriffe als eine DFA.

Der große Nachteil einer Doppelrechnung ist der zusätzliche Zeitbedarf, welcher Probleme bereiten kann. Dies betrifft vor allem Angriffe auf zeitaufwendige asymmetrische Kryptoverfahren wie RSA oder DSS. Eine weitere wirksame Abwehrmaßnahme gegen differenzielle Fehleranalysen kann dadurch bewerkstelligt werden, indem man immer nur unterschiedliche Klartexte verschlüsselt. Die einfachste Lösung ist eine dem zu verschlüsselnden Klartext vorangestellte Zufallszahl. Damit verschlüsselt der Kryptoalgorithmus immer verschiedene Daten, und eine DFA ist nicht mehr möglich.

Die Quintessenz aus Bellcore-Angriff und differenzieller Fehleranalyse lässt sich auf die Formel bringen, wonach es sich um durchaus gefährliche Angriffe handelt, die bei unzureichend ausgestatteten Chipkarten zum Erfolg führen können. Doch wurden innerhalb kurzer Zeit nach Bekanntwerden der beiden Angriffsmethoden alle Chipkarten-Betriebssysteme und Anwendungen diesbezüglich abgesichert, sodass sowohl Bellcore-Angriff als auch DFA heute keine ernsthafte Gefährdung mehr darstellen.

Angriff und Schutz: Störung des Prozessors

Ähnlich der Benutzung der differenziellen Fehleranalyse DFA beim Angriff auf die geheimen Schlüssel von Kryptoalgorithmen kann man durch Störung des Prozessors versuchen, Abläufe im Programmcode zu beeinflussen. Der sowohl bei den Herstellern von Chipkarten wie bei Chipkarten-Mikrocontrollern und einigen Systemhäusern seit ca. 1998 als Lichtangriff (*light attack*) bekannte Angriff wurde Mitte 2002 von Sergei Skorobogatov und Ross Anderson [Skorobogatov 02] als *Optical Fault Induction Attacks* veröffentlicht. Die Publikation beschreibt eine Anordnung, bei der ein handelsübliches Blitzgerät auf den Kameraadapter eines konventionellen Lichtmikroskops angeflanscht ist. Anschließend wird ein eng begrenzter Bereich des RAMs eines Standard-Mikrocontrollers (PIC16F84) angeblitzt. Mit dieser Anordnung ist es möglich, selektiv bestimmte Bits im RAM dieses – nicht gegen diese Art von Angriffen gehärteten – Mikrocontrollers in den Zustand 0 oder 1 zu setzen.

Um den Prozessor zu stören, können beispielsweise Glitches auf den Versorgungsleitungen, Lichtblitze auf den Chip oder Teile des Chips oder auch Hochfrequenz verwendet werden [Lamla 00]. Wird dann zum richtigen Zeitpunkt im Programmablauf die Störung ausgelöst, so lässt sich damit beispielsweise eine Abfrage gezielt beeinflussen. Bild 3 zeigt dazu ein einfaches Beispiel. Die dargestellte Programmfunktion hat die Aufgabe, den Inhalt eines Sendepuffers zu übertragen, dessen Grenzen von einer Anfangsadresse und einer Endadresse festgelegt sind. Schafft es der Angreifer, gezielt die Abfrage auf das Ende des Sendepuffers zu stören, werden auch die Daten an das Terminal übertragen, die im Anschluss an den Sendepuffer folgen. Befände sich nun in diesem Speicherbereich der Arbeitsspeicher für geheime Schlüssel eines Kryptoalgorithmus, ließen sich die Schlüssel mit dieser Methode unberechtigterweise auslesen.

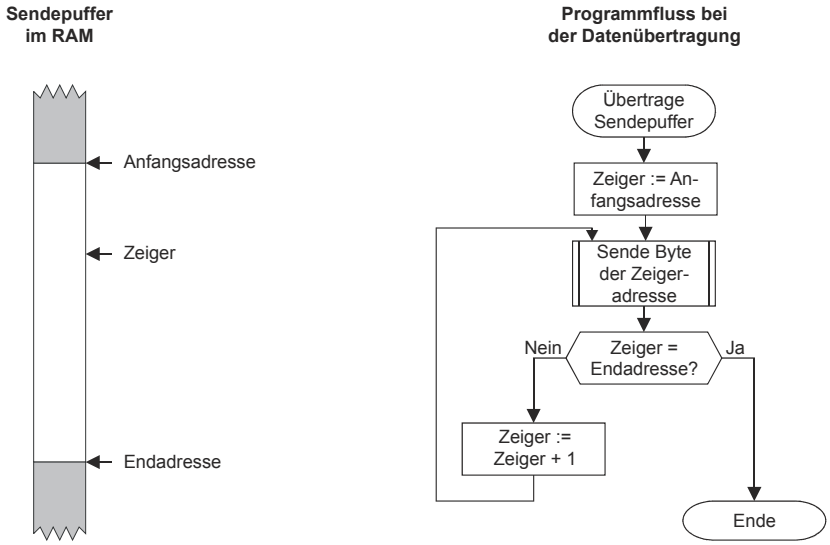

Bild 3 Beispiel für einen nicht robusten Programmablauf zum Übertragen eines Sendepuffers, der durch Störung des Prozessors mit Erfolg angegriffen werden kann.

Die Abwehr dieses Angriffs gestaltet sich vielschichtig. Wichtig ist, dass der Chipkarten-Mikrocontroller über entsprechende Sensoren verfügt, um alle Störungsversuche des Prozessors zu erkennen. Dies können auf Glitches detektierende Spannungssensoren und eine große Anzahl entsprechender Lichtsensoren auf dem Chip sein.

Die zweite Schutzebene muss in der Software realisiert werden. Der im Beispiel dargestellte Programmcode kann durch Ersatz der „=“-Abfrage durch eine „<=“-Abfrage deutlich robuster gemacht werden. Weiterhin kann als Gegenmaßnahme die Anfrage doppelt ausgeführt werden, wobei der zeitliche Abstand der beiden Abfragen eine zufällige Länge haben sollte. Damit müsste der Angreifer zwei Lichtblitze zur Manipulation der Abfrage einsetzen und stünde zudem noch vor dem Hindernis, dass er den Zeitpunkt für den zweiten Lichtblitz nicht exakt vorhersagen kann.

Zusätzlich sollten alle vertraulichen Daten im RAM unmittelbar nach ihrer Benutzung wieder gelöscht oder temporär verschlüsselt werden. Um die Auswirkungen dieses Angriffs weiter zu reduzieren, ist es ebenfalls sinnvoll, alle Geheimnisse (z. B. PIN, Schlüssel) im EEPROM zu verschlüsseln. Sollte es nämlich ein Angreifer schaffen, Teile des EEPROMs durch Manipulation von Abfragen auszulesen, so würde er als Ergebnis nur verschlüsselte Daten erhalten, die ihm nichts nützen. Steht eine MMU zur Verfügung, kann diese zusätzlich auf die Überwachung der Einhaltung von bestimmten Grenzen beim Senden von Daten konfiguriert werden. Außerdem können moderne Prozessoren illegale Maschinenbefehle und ungültige Adressen erkennen und darauf entsprechend reagieren. Dieses Abwehrszenario zeigt sehr gut, dass bei entsprechendem Zusammenspiel von Hard- und Softwareschutzmaßnahmen ein durchaus ernst zu nehmender Angriff abgeblockt werden kann.

4 Schlußfolgerung

Natürlich ist es praktisch unmöglich, ein ganzes System oder auch nur eine Chipkarte so zu bauen, dass perfekte Sicherheit herrscht, die durch nichts und niemanden gebrochen werden kann. Schließlich muss man nur den Aufwand für den Angriff genügend hoch treiben, und dann kann man in jedes System eindringen oder es manipulieren. Doch jeder potenzielle Angreifer wird, bewusst oder unbewusst, für sich und seine Ziele immer eine Art Wertanalyse machen. Denn das Ergebnis, dass er durch das Brechen eines Systems erhält,

muss die Arbeit, Geld und Zeit wert sein, die er dafür aufwendet. Falls das Resultat – sei es in monetärer Form oder auch reputationsmäßig in der (Fach-)Welt – den Aufwand nicht rechtfertigt ist, wird niemand allzu viel Energie in das Brechen eines Systems oder einer Chipkarte stecken. Dies ist eines der wesentlichen Kriterien für die Gestaltung eines sicheren systemsn mit Chipkarten.

5 Anhang

[Biham 96] Eli Biham, Adi Shamir: A new cryptoanalytic attack on DES, Internet, 1996

[Boneh 96] Dan Boneh, Richard A. DeMillo, Richard J. Lipton: On the Importance of Checking Computations, Internet, 1996

[Gandolfi 01] Karine Gandolfi, Christophe Mourtel, Francis Oliver: Electromagnetic Analysis: Concrete Results, Workshop CHES 2001, 2001

[Kocher 95] Paul C. Kocher: Timing Attacks on Implementations of Diffie-Hellmann, RSA, DSS, and Other Systems, Internet, 1995

[Kocher 98 a] Paul C. Kocher, Joshua Jaffe, Benjamin Jun: Introduction to Differential Power Analysis and Related Attacks, Internet, 1998

[Kömmerling 99] Oliver Kömmerling, Markus G. Kuhn, Design Principles for Tamper-Resistant Smartcard Processors, USENIX Workshop on Smartcard Technology, Chicago USA, 10–11 Mai 1999

[Lamla 00] Michael Lamla: Hardware Attacks on Smart Cards - Overview, Eurosmart Security Conference, Marseille, 13–15 Juni 2000

[Skorobogatov 02] Sergei Skorobogatov, Ross Anderson: Optical Fault Induction Attacks, Internet, Mai 2002