Eingebettete Sicherheit: State-of-the-art

Christof Paar, Jan Pelzl, Kai Schramm, André Weimerskirch
und Thomas Wollinger
Horst Görtz Institut für IT-Sicherheit
Ruhr-Universität Bochum, Germany
cpaar, pelzl, schramm, weika, wollinger@crypto.rub.de

Presented at D-A-CH Security 2004, University of Basel, March 30-31, 2004

Abstract

Es wird vielfach angenommen, dass die nächste Revolution in der IT-Landschaft durch die Vernetzung von eingebetteten Systemen erfolgen wird. In solche pervasiven Computeranwendungen wird IT-Sicherheit eine extrem wichtige Rolle spielen. Obwohl es starke Indikatoren gibt, dass die eingebettete Sicherheit von grosser Bedeutung sein wird, ist sie als eigenständiges Gebiet bisher kaum betrachtet worden. Ziel des vorliegenden Beitrags ist es, verschiedene Aspekt der eingebetteten Sicherheit in einer Gesamtdarstellung näher zu beleuchten. Insbesondere werden die spezifischen Probleme der eingebetteten Sicherheit näher betrachtet. Anhand von Fallbeispielen im Automobil und in ad-hoc Netzen werden zukünftige Probleme und Möglichkeiten von IT-Sicherheit in eingebetteten Anwendungen verdeutlicht. Zwei wichtige Realisierungsaspekte von sicheren eingebetteten Systemen in der Praxis, nämlich effiziente asymmetrische Verfahren in rechenbeschränkten Umgebungen und Seitenkanalattacken, werden ebenfalls diskutiert.

1 Einleitung

In den letzten Jahrzehnten hat das Internet zunehmend Computer vernetzt und hat mit Anwendungen wie dem World Wide Web und Email den Informationsfluss und die Kommunikation in vielen Lebensbereichen dramatisch beeinflusst. Geschäftsabläufe, private Kommunikation, Interaktion zwischen Bürgern und öffentlicher Verwaltung u.v.a.m. haben zum Teil revolutionäre Veränderungen erfahren. Hierdurch sind viele neue Sicherheitsprobleme, beispielhaft seien hier Anonymität, Identitätsdiebstahl, Computerviren, Schutz digitaler Inhalte etc. genannt, entstanden. Dies hat zu sehr aktiven wissenschaftlichen und kommerziellen "Communities" auf dem Gebiet der IT-Sicherheit geführt, die sich inzwischen in viele Unterdisziplinen einteilen lassen, z.B. theoretische und angewandte Kryptographie, Netzsicherheit, Computersicherheit, Digital Rights Management, Intrusion Detection. Obwohl es sicherlich noch eine Reihe offener Fragen, insbesondere bezüglich der praktischen Umsetzung von IT-Sicherheitslösungen gibt, sind viele Sicherheitsprobleme prinzipiell gelöst und es stehen zahlreiche Produkte hierfür zur Verfügung.

Eine hoch interessante Fragestellung ist nun, welche neuen Anwendungsfelder für die zukünftige IT-Sicherheit von Bedeutung sein werden. Wir glauben, dass eine Antwort hierauf der ebenso umfang- wie fassettenreiche Bereich der Sicherheit für eingebettete Systeme, im folgenden EMSEC genannt, ist. Zunächst zur Definition: Man bezeichnet ein Gerät als ein "eingebettetes System", wenn die folgenden Eigenschaften vorliegen:

- Das Gerät ist im wesentlich für eine Anwendung konzipiert (z.B. Waschmaschine, Mobiltelefon, Uhr, Automobil).
- Das Gerät ist mit "Intelligenz", d.h. mit einem Rechner, ausgestattet.
- Die Rechnerfunktionalität ist nicht sichtbar für den Benutzer, d.h. es gibt keine klassischen Computer-Benutzerschnittstellen wie Bildschirm oder Tastatur.
- Das Gerät ist nicht frei programmierbar.

Aus dieser Definition ergibt sich, dass praktisch alle (Alltags-) Geräte, die mit einem Mikroprozessor ausgestattet sind, als eingebettete Systeme zu betrachten sind. Die Bedeutung dieses Bereichs wird oft dramatisch unterschätzt, kann aber leicht an dem folgenden Fakt verdeutlicht werden: Im Jahr 2000 wurden nur 2% Prozent aller hergestellten Mikrorechner in interaktiven, d.h. herkömmlichen Computern verwandt, während die restlichen 98% in eingebetteten Anwendungen eingesetzt wurden [EGH00]. Dieses Verhältnis wird plausibel, wenn man bedenkt, dass in Automobilen des oberen Preissegments schon heute bis zu 80 Mikrorechner eingebaut sind.

Zusammenfassend kann gesagt werden, dass wir schon jetzt von eingebetteten, mit Rechnern ausgestatteten Systemen umgeben sind. Eine oft vorhergesagte Vision ist, dass die nächste IT-Revolution in der Vernetzung solcher Systeme besteht. Man spricht hier oft von Szenarien mit pervasiven (alles durchdringende) oder ubiquitären (allgegenwärtigen) Computern. Sobald dies erfolgt, wird das Thema der Absicherung solcher Systeme von großer Bedeutung sein. Wir geben im folgenden eine Reihe von Beispielen sowohl heutiger aber insbesondere auch zukünftiger eingebetteter Anwendungen, bei denen Sicherheit eine wichtige Rolle spielen wird:

- Sicherheit in Automobilen
- Sicherheit in ad-hoc Netzen
- Sicherheit im Mobilfunk
- Sicherheit in Geoinformationsanwendungen
- Sicherheit für "wearable" Computer
- Sicherheit in intelligenten Räumen
- ...

Obwohl es starke Indikatoren gibt, dass die EMSEC-Thematik von grosser Bedeutung sein wird, ist sie als eigenständiges Gebiet bisher kaum betrachtet worden. Ziel des vorliegenden Beitrages ist es, verschiedene (aber sicherlich nicht alle) Aspekte des Themas EMSEC in einer Gesamtdarstellung näher zu beleuchten. Im einzelnen werden in diesem Beitrag die folgenden Punkte behandelt. In Kapitel 2 werden die spezifischen Probleme der eingebetten Sicherheit näher betrachtet. Danach folgen zwei zukünftig wichtige Anwendungsdomänen für eingebettete Sicherheit: IT-Sicherheit in Automobilen und in ad-hoc Netzen. Ein wichtiger Aspekt der EMSEC sind Seitenkanalattacken, die in Kapitel 5 behandelt werden. In Kapitel 6 wird eine Zusammenfassung des Stands der Technik in einem weiteren wichtigen Teilaspekt der EMSEC gegeben, nämlich die Realisierung rechenintensiver asymmetrischer Verfahren auf rechenbeschränkten Mikroprozessoren. In diesem Kapitel gehen wir auch auf neue Forschungsergebnisse aus unserem Lehrstuhl ein. Abschließend werden in Kapitel 7 Anregungen für weitere Forschungen im EMSEC-Bereich gegeben.

2 Besonderheiten und Herausforderungen von Sicherheit in eingebetteten Systemen

Einführend sollte klargestellt werden, dass sich die eingebettete Sicherheit im Allgemeinen stark von der IT-Sicherheitsproblematik in Computernetzen (LAN-, Internet-, VPN-Sicherheit) unterscheidet. Die Letztere ist relativ vertraut und es stehen Lösungen wie beispielsweise Verschlüsselungssoftware, Firewalls, Intrusion Detection Systeme u.a. zur Verfügung. Wenn wir die hier vorliegende Problematik der eingebetten Sicherheit betrachten, wird deutlich, dass die genannten Lösungen aus der Welt der Computernetze zum großen Teil nicht übertragbar sind. Für eine grundlegende Einführung in das Gebiet der eingebetteten Sicherheit sei das unlängst erschienene Buch von Ross Anderson empfohlen [And01b].

Ressourcenbeschränkung: Viele der zu schützenden Systeme werden mit vergleichbar schwachen eingebetteten Prozessoren, insb. 8 oder 16 Bit Mikrocontroller ausgestattet. Für Sicherheitsanwendungen sind oft asymmetrische Algorithmen erforderlich, die extrem arithmetikintensiv sind (z.B. Berechnungen mit 1024 Bit Operanden), und die Ausführung asymmetrischer Algorithmen nicht oder nur mit sehr sorgfältigen Implementierungen erlaubt. Abschnitt 6 diskutiert diese Problematik detaillierter.

Seitenkanalattacken: Eine zentrale Komponente für die Absicherung einer IT-Anwendung sind kryptographische Algorithmen. Sowohl symmetrische als auch asymmetrische Verfahren basieren darauf, dass die zu schützende Einheit (beispielsweise ein Fahrzeugsensor, Tachometer, oder Unterhaltungselektronik) einen *geheimen* kryptographischen Schlüssel besitzt, der durch Angreifer nicht ausgelesen werden kann. Da, wie oben beschrieben, viele der potentiellen Angreifer physikalischen Zugang zu den Einheiten haben, besteht die Gefahr, dass diese durch Seitenkanalangriffe in den Besitz des Schlüssel gelangen, und damit Teile manipulieren und klonen können. Seitenkanalattacken nutzen Informationen über den Verlauf des Stromverbrauchs oder des Zeitverhaltens von kryptographischen Algorithmen aus, um den Schlüssel zu rekonstruieren. Diese Attacken wurden gegen Ende der 90er Jahre das erste Mal vorgeschlagen, und es existieren zur Zeit eine Vielzahl von Gegenmaßnahmen einerseits und verbesserten Attacken andererseits. Viele der Ergebnisse in diesem Bereich wurden in den CHES Konferenzbänden dargestellt [KP99, KP00, KNP01, KKP02].

Reverse Engineering: Verwandt mit Seitenkanalattacken sind Angriffe, die durch Methoden des Reverse Engineering versuchen in den Besitz von geheimen kryptographischen Schlüsseln zu gelangen. Hierzu gehört beispielsweise das Auslesen von Speicherzellen in eingebetteten Prozessoren oder in integrierten Schaltungen. Entsprechende Gegenmaßnahmen fallen in den Bereich des "Tamper Resistance". Fallbeispiele zu diesem Thema und den damit verbundenen Schwierigkeiten sind in [And01a] zu finden.

Beschränkte Wartungsmöglichkeiten: Im allgemeinen Fall wird es sehr schwer sein, bekannt gewordene Sicherheitsprobleme mit nachträglichen Änderungen zu verhindern. Dies ist leider aber der Alltag in der IT-Sicherheit in konventionellen Computeranwendungen: Nachdem eine neue Lücke bekannt geworden ist, werden beispielsweise Software-Patches installiert, oder der Virusscanner erhält Signaturen neuer Viren. Ein vergleichbarer Ansatz zum Beheben von Sicherheitslücken wird in den meisten eingebetteten Anwendungen per Definition nicht möglich sein, da sich eingebette Anwendungen durch ihre Nichtprogrammierbarkeit auszeichnen. Darüber hinaus werden in manchen Fällen Sicherheitsfunktionen in Hardware realisiert werden. All das wird Sicherheits-Updates in den meisten Fällen unmöglich oder nur unter großen Kosten erlauben. Dies unterstreicht die Bedeutung eines einwandfreien Security-Engineerings in der Entwurfsphase, um spätere Änderungen so klein wie möglich zu halten.

Geänderte Geschäftsmodelle: Durch die oben schon erwähnte Nichtprogrammierbarkeit von eingebetteten Anwendungen, werden die Anbieter von Sicherheitslösungen *nicht* an den Endkunden (Benutzer) verkaufen können, sondern an den Hersteller der Geräte (OEM) bzw. an Zulieferer. Diese bedeutet für die Hersteller auch, dass sie die Verantwortung für sichere Produkte übernehmen, und es nicht, wie im jetzigen PC-Markt, dem Kunden überlassen, Firewalls und Anti-Viren Software zu installieren.

3 Anwendungsbeispiel I: IT Sicherheit im Automobil

Obwohl Automobile als Anwendungsgebiet für IT-Sicherheit auf den ersten Blick *nicht* naheliegend erscheinen, gibt es starke Indikatoren dafür, dass dies eine bedeutende Domäne für eingebettete Sicherheit sein wird. Zunächst sei angemerkt, dass moderne Automobile schon sehr stark von Computer- und Kommunikationstechnik durchsetzt sind [Gre]. Beispielhaft sei hier erwähnt, dass Wagen im oberen Preissegment schon heute mit bis zu 80 Mikroprozessoren ausgestattet sind, die über verschiedene Bussysteme kommunizieren. Zum einen wird die Informationstechnik für grundlegende Fahrzeugfunktionen (Motorsteuerung, Bremsen, Lenkung) eingesetzt, daneben für Sekundärfunktionen wie Wegfahrsperre, Airbag etc. und letztlich für Anwendungen wie Telematik, online Streckenführung und in-car Infotainment.

Ein Aspekt der modernen Informationstechnik, der bisher nicht systematisch behandelt wurde, ist die Absicherung der IT-Anwendungen. Dieses Thema wird in dem gleichen Maße an Bedeutung gewinnen, in dem Automobile mit IT-Funktionalität durchsetzt werden. Spätestens mit der Kommunikationsanbindung von Fahrzeugen an externe Einheiten, z.B. über das GSM oder UMTS-Netz, wireless-LAN ("WiFi") Kanäle oder Bluetooth-Verbindungen, wird das Gefahrenpotential sprunghaft ansteigen. Wir glauben, dass das Fehlen von adäquaten Sicherheitsmaßnamen ein ernsthafter Hinderungsgrund für die Einführung zukünftiger IT-Anwendungen sein kann, die große finanzielle und technische Bedeutung in Fahrzeugen der Zukunft haben kann. Man denke hier nur an die externe Vernetzung von Fahrzeugen, welche sowohl dem Hersteller als auch dem Fahrzeugbesitzer eine große Anzahl von neuen Diensten ermöglichen wird. Trotz der Bedeutung, die IT-Sicherheit in der modernen Automobiltechnik spielen

wird, ist dieses Thema bisher kaum diskutiert worden, und die wenigen existierenden Lösungen sind zumeist ad-hoc Ansätze.

Im folgenden stellen wir eine Reihe von jetzigen und zukünftigen IT-basierten Funktionen vor, bei denen Sicherheit eine wichtige Rolle spielen wird.

Diebstahlschutz: Dies ist wahrscheinlich in Form der Wegfahrsperre die bekannteste und älteste Anwendung in der Fahrzeugtechnik, in der moderne kryptographische Methoden zum Einsatz kommen. Die kryptographischen Schwächen der ersten Versionen der Wegfahrsperre (einfaches Aufzeichnen des Codes erlaubte Klonen des Schlüssels) betonen die Wichtigkeit eines sorgfältigen Systementwurfs. Weitergehender Diebstahlschutz, z.B. von Komponenten, durch Kryptographie ist sicherlich im Bereich des Machbaren.

Schutz von Firmware: Schon heute können zahlreiche Fahrzeugcharakteristika über Firmwareupdates geändert werden. So kann bei modernen Fahrzeugen z.B. die Motorleistung durch eine geändertete Motorsteuersoftware gesteigert bzw. verringert werden. Die Absicherung solcher Updates muss aus verschiedenen Gründen geschützt werden. Zum einen will der Hersteller aus Haftungs- und Garantieüberlegungen solche Veränderungen nur kontrolliert zulassen. Zum anderen bekommt Firmware hiermit einen grossen Wert, so dass sich interessante innovative Geschäftsmodelle realisieren lassen. Mit Methoden der modernen IT-Sicherheit kann der Hersteller Kontrolle über Änderungen der Firmware erhalten.

Zugangskontrolle: Sobald Fahrzeuge in irgendeiner Form externe Kommunikation erlauben (z.B. UMTS oder Bluetooth), wird das Problem der Zugangsberechtigung akut. Man kann sich hier zahlreiche Missbrauchsszenarien vorstellen, die von dem relativ harmlosen "Stehlen" von Zustandsdaten des Fahrzeugs bis zur Manipulation des Bordcomputers oder anderer kritischer Steuergeräte reichen.

Schutz Digitaler Inhalte (Content Protection): In der Zukunft wird es zunehmend Anwendungen geben, bei denen es gilt, digitale Inhalte im Automobil zu schützen. Beispiele hierfür sind zum einen Infotainment-Inhalte Entertainment (Musik, Film) oder Kartendaten für Navigationssysteme. Hier spielt sowohl der Kopierschutz als auch Zugangsberechtigung eine Rolle.

Anonymität: Sobald eine Vernetzung des Automobils stattfindet, bei der dieses Daten sendet, ist das Problem der Verletzung der Privatsphäre zu beachten. Insbesondere bei Anwendungen wie Navigationssystemen oder anderen Geoinformationsdiensten (beispielsweise Abfrage von Restaurants in der Nähe des Fahrzeugstandortes) ist Anonymität eine wünschenswerte Eigenschaft.

Vertraulichkeit und Verlässlichkeit der Kommunikation: Ein mit der Anonymität verwandtes Problem ist die Abhörsicherheit und Verlässlichkeit der Kommunikation zwischen Automobil und der Außenwelt. Auch hier sind mannigfaltige Missbrauchsszenarien denkbar, in denen ein Angreifer beispielsweise gefälschte Telematikdaten ausgibt. Ebenso müssen Zahlungsvorgänge (Autobahngebühr!) gegen Abhören und Verfälschung gesichert sein.

Rechtliche Zwänge: Ein weiteres Anwendungsgebiet moderner IT-Sicherheit sind solche Situationen, in denen der Gesetzgeber gewisse IT-Funktionen vorschreibt. Beispiele sind z.B. die elektronischen Fahrtenschreiber in LKWs, die die Einhaltung von Ruhezeiten sicherstellen. Solche Systeme müssen gegen Manipulationen geschützt sein, und Signaturgesetze erlauben IT-basierte Lösungen, die juristisch durchsetzbar sind.

4 Anwendungsbeispiel II: Sicherheit in ad-hoc und pervasiven Netzen

Wie schon gesagt ist zu erwarten, dass in der Zukunft ein Mikrochip in nahezu alle Geräte wie Kaffeemaschinen, Thermostate sowie Radiowecker eingebaut werden wird. Werden diese Geräte mit einer drahtlosen Kommunikation ausgestattet, so könnte zusammen mit schon vorhandenen Rechnern wie Mobiltelefonen und PCs ein extrem weitverzweigtes drahtloses Netzwerk entstehen. Da diesem Netzwerk Geräte kontinuierlich hinzugefügt oder entfernt werden können, sollte dieses Netzwerk selbstorganisierend sein, und nicht auf eine zentrale Infrastruktur angewiesen sein. Jeder Knoten in dem Netzwerk verlässt sich dann auf seine Nachbarn indem er diesen seine Dienste anbietet und deren Dienste beansprucht,

z.B. bei der Weiterleitung und dem Senden von Datenpaketen. Solch ein Netzwerk wird Ad-hoc Netz genannt. Hierbei existiert kein einzelne Fehlerquelle oder Angriffspunkt, da es keine zentralen Server gibt. In vielen Fällen ist zu erwarten, dass Basisstationen den Zugang zum Internet ermöglichen.

Anwendungen Typische Beispiele von Ad-hoc Netzen sind Bluetooth, HiperLAN2, und eingeschränkt auch WLAN (802.11). Ad-hoc Netze wurden ursprünglich in den 70'er Jahren im Rahmen militärischer Forschung (DARPA) entwickelt. Daher werden Anwendungen auch häufig in militärischen Bereichen oder in Katastrophengebieten gesehen, z.B. bei der Kommunikation und Überwachung von Soldaten, oder dem Aufbau eines ausgefallenen Telefonnetzwerks. Weitere Beispiele sind ein Netz mobiler Telefone mit Hilfe von Bluetooth, um kostenlos telefonieren zu können, und Sensornetzwerke. Einen ersten Eindruck winziger Mikrochips, die als Sensoren wie Temperatur- oder Lichtfühler in solch einem Sensornetzwerk fungieren können, geben die Smartdust Geräte, die an der Berkeley University entwickelt werden. Schon nahezu alltagstauglich sind sogenannte Radio Frequency Identification (RFID) Etiketten. Dies sind kleinste passive Elemente, die einen kurzen Bitstring speichern können. So ist es vorstellbar, dass RFID Etiketten bald die Barcodes ablösen werden. Dann müsste an der Supermarktkasse nicht mehr jedes Produkt aus dem Einkaufswagen genommen werden, um den Endpreis festzustellen, sondern alle Waren könnten gleichzeitig erfasst werden, wenn der Einkaufwagen durch die Kasse geschoben wird. Diese Technologie kann zu riesigen Einsparungen in allen Bereichen der Logistik führen. Weiterhin ist vorstellbar, dass Tablettenpackungen ein RFID Etikett beinhalten. Dadurch kann der Anwender automatisch vor Nebenwirkungen gewarnt werden, die durch die Einnahme verschiedener nicht-verträglicher Tabletten auftreten könnten.

Sicherheit Wegen der Beschränkungen der mobilen Geräte, insbesondere bezüglich Rechenleistung, Speicherkapazität und Batterieleistung, müssen die Sicherheitsanforderungen in Ad-hoc Netzen anders betrachtet werden als in statischen Netzwerken. So ist es zum Beispiel nicht möglich, jedes Datenpaket zum Zweck des sicheren Routings zu signieren, um den Ursprung sicherzustellen. Dies scheitert aufgrund der beschränkten Rechenleistung und der Notwendigkeit einer Public-Key Infrastruktur (PKI), die sich im Grundsatz nicht mit dem dezentralen Charakter eines Ad-hoc Netzes verträgt. Das sichere Routing von Datenpaketen ist ein wichtiges, aber bisher nicht grundsätzlich gelöstes Forschungsthema. Da in einem Ad-hoc Netzwerk jeder Knoten potentiell ein Router ist, müssen sichere und robuste Verfahren eingesetzt werden. Mögliche Lösungsansätze werden z.B. in [BH01, LPW03, WW03] behandelt. Aufgrund der mangelnden physikalischen Sicherheit ist es in Ad-hoc Netzen zudem wichtig, dass einige bösartige Knoten die Funktionsweise des Netzes nicht gefährden.

Die Gefahren aufgrund mangelnder Sicherheit wird insbesondere in Szenarien sichtbar, in denen Alltagsgegenstände mit Sensoren ausgestattet sind. So können die oben erwähnten RFID Etiketten nicht nur Logistikabläufe beschleunigen, sondern sie können auch zur Erstellung von Kundenprofilen genutzt werden. Sie können aber auch eine echte Gefahr bedeuten. Es gibt Überlegungen, winzige RFID Etiketten in Geldscheinen einzubetten, um diese fälschungssicher zu gestalten. Dies bedeutet allerdings auch, dass ein Taschendieb mit einem einfachen Gerät sofort feststellen kann, wer große Geldmengen bei sich trägt. Eine mangelnde Sicherheit gefährdet also die Privatsphäre, und kann auch zu weiterem Schaden führen.

Lösungsansätze Wir betrachten nachfolgend mögliche Lösungsansätze, indem wir eine Kategorisierung der möglichen Szenarien angeben.

Militärische Anwendungen: Es gibt nur eine Authorität, der alle Knoten untergestellt sind. Weiterhin sind Kostenfragen hier weniger bedeutend. Daher sind Lösungen mit symmetrischer Kryptographie vorstellbar, oder auch Public-key Techniken. Da es nur eine Authorität gibt, könnte der Ansatz einer verteilten PKI gewählt werden [ZH99]. Hierbei wird die Aufgabe der Erstellung und Erneuerung von Zertifikaten auf viele Knoten verteilt, so dass eine Manipulation erschwert wird.

Heimanwendungen: Auch hier gibt es nur eine Authorität, nämlich den Besitzer der Heimgeräte. Daher können symmetrische Kryptographiemethoden genutzt werden. So ist z.B. ein Schlüsselaustausch als Eingabe einer PIN möglich. Eleganter ist die Möglichkeit des Schlüsselaustauschs durch einen physikalischen Kontakt, wie es z.B. als Resurrecting Duckling durch Stajano vorgeschlagen wird [SA99].

Meeting: Wenn sich eine kleine Personengruppe trifft, die gegenseitig auf Dienste zugreifen will, bietet sich ein passwortbasierter Schlüsselaustausch an. Dabei wird ein Gruppenpasswort verteilt, also das Vertrauen der Leute ineinander auf das Ad-hoc Netz abgebildet. Bei Gruppen ohne Vertrauensbasis funktioniert dieser Ansatz nicht.

Heterogenes Pervasives Netzwerk: Hierbei besteht das Netzwerk aus heterogenen Knoten, die von einer Vielfalt an Authoritäten betrieben werden. Falls das Ad-hoc Netzwerk eine Verbindung zum Internet hat, können Sicherheitsmechanismen wie z.B. ein Kerberos Server genutzt werden. Andernfalls ist es notwendig, ein Vertrauensnetz aufzubauen. Dabei muss jeder Knoten mit der Zeit ein Vertrauensnetz zu anderen Knoten aufbauen. Dies findet in ähnlicher Form Anwendung im Web of Trust von PGP.

Sensornetzwerke In Sensornetzwerken ist die Benutzung asymmetrischer Kryptographie aufgrund der beschränkten Rechenleistung der Sensoren nahezu ausgeschlossen, jedoch ist die Gefahr physikalischer Angriffe sehr hoch, so dass eine einfache symmetrische Lösung unzureichend ist. Wir haben jedoch gezeigt, dass ein vernünftiger Sicherheitsansatz mit rein symmetrischer Kryptographie in Sensornetzwerken möglich ist, der auch auf extrem leistungsschwachen Mikrochips realisiert werden kann [WW].

Abschliessend ist hier zu sagen, dass keine einheitliche Lösung existiert, die Sicherheit in Ad-hoc Netzwerken sicherstellt. Bei der Erstellung von Lösungen müssen die Anforderungen und Annahmen gründlich gewählt werden. Dies bedeutet meist, dass Ansätze größtenteils auf symmetrischer Kryptographie basieren sollten, um teure Rechenoperationen zu vermeiden. Weiterhin muss natürlich aufgrund der großen Anzahl an Knoten eines pervasiven Netzwerkes immer davon ausgegangen werden, dass ein Teil der Knoten infiltriert und manipuliert werden kann. Dies sollte jedoch keinen merkbaren Einfluß auf das Gesamtnetzwerk haben.

5 Physikalische Attacken

Kryptographische Algorithmen wurden bis vor wenigen Jahren unter der Annahme entwickelt, dass deren physikalische Implementierungen einer "Black Box" gleichen, welche keine Informationen über zu Grunde liegende Verarbeitungsvorgänge preisgibt. Mitte der Neunzigerjahre wurden jedoch die ersten physikalischen Attacken bekannt, welche prinzipiell in zwei Gruppen klassifiziert werden können.

- Invasive Attacken basieren auf dem Entfernen der Passivierungsschicht eines Prozessors mit entsprechenden Ätzwerkzeugen aus der Mikroelektronik [And01b, KK99]. Liegt der zu untersuchende Chip erst einmal frei, ist es möglich mittels einer *probing station* den Datenbus, Adressbus oder auch Speicherzellen zu beobachten und deren Zustand zu protokollieren. Als Maßnahme gegen invasive Attacken sind Smartcard Mikroprozessoren der neusten Generation oft mit Schutzsensorgittern, wie z.B. einem *active shield* ausgerüstet. Dabei handelt es sich in der Regel um ein Sensorgitter in der obersten Metallisierungsebene, durch welches kontinuierliche Stromsignale fließen. Manipulationsversuche des Sensorgitters werden von dem Prozessor erkannt, so dass definierte Aktionen (z.B. das Löschen von sensitiven Daten, Endlosschleife) eingeleitet werden [Sei04].
- Nichtinvasive Attacken basieren auf der Tatsache, dass Ausführungsdauer, Stromverbrauch sowie elektromagnetische Abstrahlung direkt mit den ausgeführten Rechenoperationen bzw. den verarbeiteten Operanden korrelieren können. Paul Kocher et al. demonstrierten im Jahr 1998, dass praktisch alle zu diesem Zeitpunkt erhältlichen Smartcards durch Seitenkanalattacken gebrochen werden konnten [KJJ98, KJJ99]. Gegenwärtig enthalten die meisten sicherheitskritischen Smartcardprozessoren auf Hard- bzw. Software basierende Maßnahmen, um Seitenkanalattacken entgegenzuwirken.

An unserem Lehstuhl besteht die Möglichkeit Stromprofilattacken, also Seitenkanalattacken, welche den Stromverbrauch eines Prozessors analysieren, durchzuführen. Der Aufwand ist im Vegleich zu anderen physikalischen Attacken gering, denn in der Praxis reicht dazu ein digitales Speicheroszilloskop und ein Widerstand. Zunächst wird ein geringer Reihenwiderstand (typischerweise zwischen 10Ω und 50Ω) zwischen dem Massekontakt des Prozessors und der externen Masse der Spannungsquelle geschaltet. Dann wird mit einem digitalen Speicheroszilloskop die über dem Widerstand abfallende Spannung gemessen, welche proportional zum Laststrom ist. Darüber hinaus wird die Spannungsquelle des Prozessor in der Regel durch eine möglichst rauscharme Spannungsquelle ersetzt, um eingestreute Messfehler zu minimieren.

Die beiden bekanntesten Stromprofilattacken sind die einfache Stromprofilanalyse (engl. *Simple Power Analysis (SPA)*) und die differentielle Stromprofilanalyse (engl. *Differential Power Analysis (DPA)*) [KJJ98, KJJ99]. Bei der SPA analysiert ein potenzieller Angreifer das Stromprofil zu einem festen Zeitpunkt t unter der Annahme, dass ein verarbeiteter Operand direkt von dem geheimen Schlüssel abhängt und

mit dem Stromverbrauch zu diesem Zeitpunkt korreliert. In der wissenschaftlichen Literatur wurde mehrfach gezeigt, dass insbesondere das Hamming Gewicht der verarbeitetenden Operanden sowie Carry Flag abhängige Verzweigungs- und Rotationsinstruktionen mit Hilfe einer einfachen Stromprofilanalyse erkannt werden können [MDS99, MS00]. Eine SPA setzt allerdings detaillierte Kenntnisse der zu untersuchenden Hardware sowie des implementierten Algorithmus voraus und ist daher in der Praxis nicht immer durchführbar.

Die differentielle Stromprofilanalyse DPA basiert auf einer statistischen Auswertung der gemessenen Stromkurven. Dazu stellt ein Angreifer zunächst eine Hypothese bzgl. des geheimen Schlüssels auf. Im Falle des *Data Encryption Standard (DES)* lässt sich beispielsweise während einer Verschlüsselung der Ausgang einer S-Box in der ersten Runde vorhersagen, falls der Angreifer eine korrekte 6-Bit Schlüsselhypothese aufstellt und den Klartext kennt. Verschlüsselt der Angreifer n randomisierte Klartexte, so lassen sich die entsprechenden Stromprofilmessungen für jede Schlüsselhypothese in zwei Summensignale aufaddieren: ein Summensignal enthält alle Messungen, für die ein fest gewähltes Ausgangsbit der S-Box den Wert 0 hat, das andere Summensignal enthält alle Messungen, für die das Ausgangsbit den Wert 1 hat. Da die Klartexte zufällig generiert werden und die Ausgangswerte der S-Boxen gleichverteilt sind, werden in jedem Summensignal ca. $n/2$ Stromkurven aufaddiert. Als nächstes untersucht der Angreifer das Differenzsignal der beiden Summensignale. Bei einer falschen Schlüsselhypothese sind die aufsummierten Messkurven zu allen Zeitpunkten aufgrund der randomisierten Operanden unkorreliert, das Differenzsignal ist daher verschwindend gering und strebt gegen Null. Bei einer korrekten Schlüsselhypothese korrelieren die aufsummierten Stromprofile genau in den Instruktionen, welche den Ausgang des untersuchten S-Box Bits verwenden. In dem Differenzsignal ist dies an deutlichen Ausschlägen zu erkennen. Im Vergleich zur SPA bietet die DPA zwei deutliche Vorteile: der Angreifer benötigt weder genaue Kenntnisse über die eingesetzte Hardware noch Details über die jeweilige Softwareimplementierung.

Mittlerweile gibt es zahlreiche hard- und softwarebasierte Schutzmaßnahmen gegen Seitenkanalattacken. Typische Softwaremaßnahmen sind z.B. das randomisierte Maskieren von Operanden und die Verwendung von maskierten S-Boxen [GP99, CCD00]. Smartcard Hersteller verwenden mittlerweile häufig Hardware Gegenmaßnahmen, wie z.B. zufällig generierte *wait states*, welche die Summensignale der DPA desynchronisieren, Rauschgeneratoren sowie DPA-resistente Gatterlogikfamilien [Sei04].

An unserem Institut wurde darüber hinaus eine neue Klasse von Seitenkanalattacken entwickelt, welche interne, schlüsselabhängige Kollisionen in kryptographischen Algorithmen, wie z.B. DES, AES und IDEA detektiert um den geheimen Schlüssel zu rekonstruieren [SWP03]. Kollisionsattacken zeichnen sich im Vergleich zur DPA aufgrund des Geburtstagsparadoxon durch einen geringen Meßaufwand aus. Darüber hinaus wirken sich Kollisionen in der Regel auf mehrere Instruktionen aus[1] und sind in folgedessen leicht zu detektieren.

In der Kryptographie bezeichnet der Ausdruck Kollision den Fall, dass der Ausgangswert einer Funktion trotz Änderung des Eingangswertes unverändert bleibt. Eine Klasse von Funktionen, die diese Eigenschaft uneingeschränkt erfüllt, sind z.B. nicht-injektive Funktionen. Eine partielle Kollision bezeichnet den Fall, dass nur ein bestimmter Teil des Ausgangswertes einer Funktion kollidiert. So lassen sich im Fall des AES partielle, schlüsselabhängige Kollisionen in den einzelnen Bytes des 32-Bit Ausgangswerts der *Mix Column* Transformation der ersten Runde erzeugen [SLFP04]. Wird darüber hinaus das Geburtstagsparadoxon ausgenutzt, um möglichst schnell, d.h. mit einer geringen Anzahl Verschlüsselungen und in Folge dessen mit geringem Meßaufwand, Kollisionen in den Ausgangsbytes der *Mix Column* zu detektieren, so ist es möglich, mit **nur** 40 Verschlüsselungen Kollisionen in allen 16 Ausgangsbytes der *Mix Column* zu verursachen. Unter der Verwendung vorberechneter Tabellen ist es schliesslich möglich, den geheimen 128-Bit Schlüssel zu bestimmen [SLFP04].

6 Asymmetrische Verfahren in eingebetteten Umgebungen

Es gibt viele Forschungsanstrengungen auf dem Gebiet der effizienten Algorithmen für kryptographische Anwendungen. Leider gibt es wenige Publikationen, in welchen die schnelle Implementierung von Kryptosystemen auf speziellen Plattformen wie z.B. eingebetteten Prozessoren untersucht wird. Diese kommen

[1] im Fall des DES kollidiert beispielsweise beinahe die gesamte zweite Runde

oft in kostensensitiven Applikationen zum Einsatz, in denen neben dem geringen Preis des Prozessors oftmals auch ein minimaler Energieverbrauch wünschenswert ist. Demgegenüber steht die durch die Anwendung einzuhaltende maximale Ausführungszeit des zu implementierenden Algorithmus. Die zentrale Aufgabe ist es daher, einen bestmöglichen Kompromiss zwischen allen genannten Eigenschaften zu finden.

Zur Realisierung von asymmetrischen Algorithmen bieten sich kryptographische Einwegfunktionen an, deren Sicherheit auf dem Faktorisierungsproblem (RSA) oder auf dem Lösen des diskreten Logarithmus (DL) Problems beruhen. Letztere lassen sich auf Gruppen von elliptischen und hyperelliptischen Kurven übertragen (ECC, HECC). RSA und DL sind seit mehr als 20 Jahren die meistverwendeten asymmetrischen Algorithmen, werden aber zusehends von neueren, auf elliptischen Kurven basierenden Verfahren abgelöst. Der Grund hierfür liegt in der Komplexität von RSA und DL gegenüber ECC: für gängige Sicherheitsniveaus sind RSA und DL Operanden 1024 Bit groß, wohingegen ECC mit Operanden um 160 Bit die gleiche Sicherheit bieten. Hyperelliptische Kurven (HEC) erlauben sogar noch geringere Bitlängen bei gleichbleibender Sicherheit. HECC werden zur Zeit intensiv erforscht und in den letzten Veröffentlichungen [PWGP03, PWP03a, PWP03b] wurde erstmals gezeigt, dass sie so effizient wie ECC sind. Nachfolgend nun ein knapper Abriss an relevanten Forschungsergebnissen zum Thema effiziente asymmetrische Algorithmen auf eingebetteten Prozessoren.

In [Bar86] wird die Implementierung von 512-Bit RSA auf einem DSP (TI-TMS32010) vorgestellt. Eine Exponentiation dauert im Durchschnitt 2.6 Sekunden bei einer Taktfrequenz von 20 MHz. [DK90] beschreibt eine RSA-Realisierung auf einem Motorola DSP56000 mit 20 MHz. Unter Verwendung des Chinesischem Restwertsatzes (CRS) wird ein Datendurchsatz von 11.6 Kbits/s erreicht. Ohne CRS erreicht die Implementierung einen Durchsatz von 4.6 Kbits/s.

Die Veröffentlichung [HNM98] schildert die Umsetzung von ECDSA über endlichen Körpern $GF(p)$ auf einem 16-Bit Mikrocomputer (M16C) mit 10MHz. Unter Verwendung eines festen Punktes kann die sehr spezielle Implementierung durch Vorausberechnungen eine ECDSA Signatur in 150 msec erzeugen. Wird ein zufälliger Punkt der Kurve gewählt, dauert das Erstellen der Signatur 480 msec. Die Verifikation nimmt in beiden Fällen 630 msec in Anspruch.

Der Beitrag [ITT+99] stellt mehrere Methoden zur Realisierung asymmetrischer Algorithmen auf einem (rechenstarken) DSP (TI-TMS320C6201,200 MHz) vor. Eine 1024-bit RSA Signatur kann in 11,7 ms und eine Verifikation in 1,2 ms erreicht werden. 192-bit ECDSA Signaturen dauern 1.67 ms, Verifikationen 6.28 ms.

Im Gegensatz zu den leistungsstarken Signalprozessoren sind Implementierungen auf 8 Bit Mikrocontrollern eine wahre Herausforderung, akzeptable Laufzeiten zu erreichen. Die besten Zeit auf dem (stark verbreiteten) 8051 Mikroprozessor ohne Coprozessor wird in [WBP00] beschrieben. Es werden Zeiten von 1,95 s für eine 134-bit Punktmultiplikation mit Vorausberechnungen und 8,37 s für die gleiche Operation mit einen zufälligen Punkt erreicht.

Der Motorola Dragonball mit 16 MHz (populär in Palmpilots u.ä.) zählt zu den Prozessoren im mittleren Leistungssegment. [WPS01] stellt eine Implementierung von ECC unter Verwendung von sog. Koblitzkurven auf dem Dragonball vor. Eine ECDSA Signatur ist in weniger als 0,9 s und eine Verifikation in weniger als 2,4 s möglich. [GBKP01] beschreibt eine weiter low-cost EC Implementierung auf einem 16-bit DSP (TI-MSP430x33x) mit einer Taktgeschwindigkeit von 1 MHz. Den Forschern ist es gelungen, auf diesem Mikrokontroller eine EC Punktmultiplikation in 3.4 Sekunden ohne Vorberechnung durchzuführen.

Spezielle Erweiterungskörper, sogenannte *Optimal Extension Fields* (OEF), ermöglichen durch Anpassung der Körperstruktur an die Wortgröße des Prozessors schnelle Körperarithmetik. Die Referenz [CSL00] berichtet von einer ECC Implementierung auf einem 8-Bit Prozessor über einem Körper mit 160 Bit. Es wird eine Laufzeit von 122 msec für eine 160-bit Punktmultiplikation auf einem CalmRISC bei 20 MHz (mit Coprozessor) erreicht.

Hyperelliptische Kurven sind potentiell noch besser für eingebettete Anwendungen geeignet als elliptische Kurven, da Berechnungen mit Zahlen erfolgen können, die nur 40–80 Bit lang sind. Der Nachteil von HEC liegt in der gegenüber EC erheblich komplexeren Arithmetik, d.h. es müssen für eine Gruppenoperation mehr Berechnungen über dem Grundkörper durchgeführt werden. Aus diesem Grund hielt man bisher das hyperelliptische Kryptosystem dem elliptischen Kryptosystem in punkto Geschwindigkeit für unterlegen.

Uns ist es gelungen, die Anzahl der Berechnungen für spezielle HECC um über 50% des zuvor bekannten Wertes zu senken.

In [WPW+03] haben wir eine intensive Analyse verschiedener HECC Algorithmen auf mehreren, derzeit für die Praxis relevanten eingebetteten Prozessoren durchgeführt. Die Dauer einer Publik-Key Operation (Gruppenordnung $\approx 2^{160}$) beträgt auf dem PowerPC@50MHz 84.9 msec, auf dem ARM7@50MHz 316.6 msec. und auf dem ColdFire@90MHz 123.6 msec. Die Implementierungen dieser Arbeit verdeutlichen die praktische Relevanz hyperelliptischer Kryptosysteme. Außerdem wurde der Einfluss der Prozessorarchitektur (CPU, Cache, RAM etc.) analysiert. Im Falle des PowerPCs konnte der Durchsatz durch die Benutzung von Cache um einen Faktor 8 gesteigert werden. Desweiteren wurde in [WPW+03] zwei unterschiedliche Arten der Implementierung vorgestellt: für beliebige und spezielle Eingabeparameter (Kurven, Körper usw.). Die Erste Variante ist um 50% ineffizienter als die spezielle Implementierung, hat aber Vorteile in einer flexiblen Umgebung.

Tabelle 1 zeigt einen Teil unserer aktuellen Ergebnisse für die Laufzeiten einer Skalarmultiplikation mit HECC auf einem ARM7 Prozessor [PWP03a, PWGP03].

Table 1: Hyperelliptische Kurven auf dem ARM7TDMI@80MHz

Geschlecht	Körper	Gruppenordnung	Add. [μs]	Verdoppel. [μs]	Punktmult. [ms]
4	$\mathbb{F}_{2^{40}}$	2^{160}	1315	740	172.43
3	$\mathbb{F}_{2^{54}}$	2^{162}	615	212	61.45
2	$\mathbb{F}_{2^{81}}$	2^{162}	471	296	69.06

Unsere Untersuchungen zeigen weiterhin, dass hyperelliptische Kurven bei geeigneter Parameterwahl elliptischen Kurven überlegen sein können. In [PWGP03] stellen wir eine eine Metrik vor, um im Vorfeld einer Implementierung die relative Effizienz der verschiedenen Algorithmen auf speziellen Prozessoren abschätzen zu können.

7 Zusammenfassung

Bisher stellte die zunehmende Vernetzung von Computern ein gravierendes Sicherheitsproblem für Firmen und Organisationen dar. Durch den massiven Einsatz eingebetteter Kleinstcomputer überträgt sich diese Problematik in alle Bereiche des alltäglichen Lebens. Das dadurch entstehende Bedrohungspotenzial wird z.Zt. in Industrie und Forschung unterschätzt.

Im folgenden werden eine Reihe offener Fragen im Bereich der eingebetteten Sicherheit aufgelistet:

- In vielen eingebetteten Anwendungen unterscheiden sich die Bedrohungsszenarien stark von konventineller Internetsicherheit. Dies kann zu grundsätzlich anderen Annahmen und Systemkonzepten führen. Eine saubere Untersuchung der Angriffspotentiale für ausgesuchte eingebettete Anwendungsdomänen erscheint hier lohnenswert.
- Resistenz gegen Seitenkanalattacken: In den letzten Jahren, hatte sich die Forschung hauptsächlich mit softwarebasierten Gegenmaßnahmen beschäftigt (siehe [KP99, KP00, KNP01, KKP02]). Wichtige Beiträge könnten hier im Bereich der Hardwaregegenmaßnahmen und der automatischen Generierung der Gegenmaßnahmen durch Entwurfswerkezuge geleistet werden.
- Symmetrische ultra low-cost Algorithmen: Der Einsatz von symmetrischen Algorithmen in eingebetteten Systemen ist wichtig für die Authentisierung und Verschlüsselung. Für Systeme mit stark limitierten Ressourcen (z.B. RFID Tags) wäre es notwendig, Algorithmen zu entwickeln, welche weniger als 1000 Gatter für die Realisierung benötigen. Vereinzelte Algorithmen gibt es bereits, deren theoretische Grundlagen jedoch noch weitgehend unerforscht sind.
- Public-key Kryptographie: Flaschenhals jeder Realisierung von kryptographischen Protokollen sind Public-Key Algorithmen. Für Anwendungen in einer pervasiven Umgebung sollte die Komplexität dieser Algorithmengruppe um mindestens eine Größenordung gesenkt werden.
- Protokolle für ad-hoc Netze: Die klassischen kryptographischen Protokolle sind wenig geeignet für die Anforderungen in ad-hoc Netzwerken. Obwohl es schon eine Reihe Protokolle für ad-hoc Netze gibt, ist bisher sehr wenig bezüglich Praxistauglichkeit getestet worden.

- Low-cost Tamper Resistance: Die bestehenden Maßnahmen gegen physikalische Angriffe (Tamper Resistance) sind aus Kostengründen nicht ohne weiteres auf eingebettete low-cost Chips möglich.

References

[And01a] R. Anderson. Protecting embedded systems — the next ten years. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2001*, volume LNCS 2162, pages 1–2. Springer-Verlag, 2001. Invited Talk.

[And01b] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley and Sons, 2001.

[Bar86] P. Barrett. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume LNCS 263, pages 311–323, Berlin, Germany, August 1986. Springer-Verlag.

[BH01] L. Buttyán and J.-P. Hubaux. Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. In *Technical Report DSC/2001/001,Swiss Federal Institute of Technology – Lausanne, Department of Communication Systems*, 2001.

[CCD00] C. Clavier, J.S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 252–263. Springer-Verlag, 2000.

[CSL00] Jae Wook Chung, Sang Gyoo Sim, and Pil Joong Lee. Fast Implementation of Elliptic Curve Defined over $GF(p^m)$ on CalmRISC with MAC2424 Coprocessor. In Çetin K. Koç and Christof Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2000*, pages 57–70, Berlin, 2000. Springer-Verlag.

[DK90] S. R. Dussé and B. S. Kaliski. A Cryptographic Library for the Motorola DSP56000. In I. B. Damgård, editor, *Advances in Cryptology — EUROCRYPT '90*, volume LNCS 473, pages 230–244, Berlin, Germany, May 1990. Springer-Verlag.

[EGH00] D. Estrin, R. Govindan, and J. Heidemann. Embedding the Internet. *Communications of the ACM*, 43(5):39–41, May 2000.

[GBKP01] J. Guajardo, R. Bluemel, U. Krieger, and C. Paar. Efficient Implementation of Elliptic Curve Cryptosystems on the TI MSP430x33x Family of Microcontrollers. In K. Kim, editor, *Fourth International Workshop on Practice and Theory in Public Key Cryptography - PKC 2001*, volume LNCS 1992, pages 365–382, Berlin, February 13-15 2001. Springer-Verlag.

[GP99] L. Goubin and J. Patarin. DES and differential power analysis: the duplication method. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 1999*, volume LNCS 1717, pages 158–172. Springer-Verlag, 1999.

[Gre] D. Grell. Computer im Auto. c't 14/2003, S. 170.

[HNM98] Toshio Hasegawa, Junko Nakajima, and Mitsuru Matsui. A Practical Implementation of Elliptic Curve Cryptosystems over $GF(p)$ on a 16-bit Microcomputer. In Hideki Imai and Yuliang Zheng, editors, *First International Workshop on Practice and Theory in Public Key Cryptography — PKC'98*, volume LNCS 1431, pages 182–194, Berlin, 1998. Springer-Verlag.

[ITT+99] K. Itoh, M. Takenaka, N. Torii, S. Temma, and Y. Kurihara. Fast Implemenation of Public-Key Cryptography on a DSP TMS320C6201. In Çetin K. Koç and Christof Paar, editors, *Proceedings of the First Workshop on Cryptographic Hardware and Embedded Systems — CHES'99*, volume LNCS 1717, pages 61–72, Berlin, Germany, August 1999. Springer-Verlag.

[KJJ98] P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. http://www.cryptography.com/dpa/technical, 1998. Manuscript, Cryptography Research, Inc.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.

[KK99] O. Kommerling and M. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Workshop on Smartcard Technology (Smartcard '99)*, pages 9 – 20. USENIX Association, May 1999.

[KKP02] B. S. Kaliski, Jr., Ç. K. Koç, and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2002*, volume LNCS 2523, Berlin, Germany, August 13-15, 2002. Springer-Verlag.

[KNP01] Ç. K. Koç, D. Naccache, and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2001*, volume LNCS 2162, Berlin, Germany, May 13-16, 2001. Springer-Verlag.

[KP99] Ç. K. Koç and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems — CHES'99*, volume LNCS 1717, Berlin, Germany, August 12-13, 1999. Springer-Verlag.

[KP00] Ç. K. Koç and C. Paar, editors. *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, Berlin, Germany, August 17-18, 2000. Springer-Verlag.

[LPW03] B. Lamparter, K. Paul, and D. Westhoff. Charging support for ad hoc stub networks. 2003. Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications.

[MDS99] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[MS00] R. Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smart Cards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 78 – 92. Springer-Verlag, 2000.

[PWGP03] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar. Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves. In Ç. K. Koç and C. Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2003*. Springer-Verlag, 2003.

[PWP03a] J. Pelzl, T. Wollinger, and C. Paar. Low Cost Security: Explicit Formulae for Genus-4 Hyperelliptic Curves. In *Tenth Annual Workshop on Selected Areas in Cryptography — SAC 2003*. Springer-Verlag, 2003.

[PWP03b] Jan Pelzl, Thomas Wollinger, and Christof Paar. High performance arithmetic for hyperelliptic curve cryptosystems of genus two. Cryptology ePrint Archive, Report 2003/212, 2003. http://eprint.iacr.org/.

[SA99] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *The 7th International Workshop on Security Protocols*. Springer-Verlag, 1999. LNCS 1796.

[Sei04] J.-P. Seiffert. CHES 2004 - Rump Session, September 2004. Smartcard Security and Hardware Countermeasures.

[SLFP04] K. Schramm, G. Leander, P. Felke, and C. Paar. A Collision-Attack on AES Combining Sidechannel- and Differential-Attack. In *(submitted to Eurocrypt 2004)*, 2004.

[SWP03] K. Schramm, T. Wollinger, and C. Paar. A New Class of Collision Attacks and its Application to DES. In Thomas Johansson, editor, *Fast Software Encryption — FSE '03*, volume LNCS 2887, pages 206 – 222. Springer-Verlag, February 2003.

[WBP00] A. Woodbury, D. V. Bailey, and C. Paar. Elliptic curve cryptography on smart cards without coprocessors. In *IFIP CARDIS 2000, Fourth Smart Card Research and Advanced Application Conference*, Bristol, UK, September 20–22, 2000. Kluwer.

[WPS01] A. Weimerskirch, C. Paar, and S. Chang Shantz. Elliptic Curve Cryptography on a Palm OS Device. In V. Varadharajan and Y. Mu, editors, *The 6th Australasian Conference on Information Security and Privacy — ACISP 2001*, volume LNCS 2119, pages 502–513, Berlin, 2001. Springer-Verlag.

[WPW+03] T. Wollinger, J. Pelzl, V. Wittelsberger, C Paar, G. Saldamli, and Ç. K. Koç. Elliptic & hyperelliptic curves on embedded μp. *ACM Transactions in Embedded Computing Systems (TECS)*, 2003. Special Issue on Embedded Systems and Security.

[WW] A. Weimerskirch and D. Westhoff. Zero Common-Knowledge Authentication for Pervasive Networks. In *Selected Areas in Cryptography - SAC, 2003*.

[WW03] A. Weimerskirch and D. Westhoff. Identity Certified Authentication for Ad-hoc Networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, 2003.

[ZH99] L. Zhou and Z.J. Haas. Securing Ad Hoc Networks. 13(6), 1999.