B 10 – Sicherheit von Informationssystemen
Wolfgang Semar, Sascha Beck

1. Anforderungen an die Sicherheit von Informations- und Kommunikationssystemen

Die Informatisierung insbesondere in Form von weltweiten, offenen Computernetzen setzt hohe Ansprüche an die sichere Übermittlung und Speicherung von Daten. Die Art und vor allem die Menge der Daten, die schützenswert sind, haben sich mit der Verbreitung der elektronischen Datenverarbeitung vom ursprünglich militärischen über das geschäftliche bis hin zum privaten Umfeld gewandelt. Für alle Partizipanten spielt die Sicherheit und die Verlässlichkeit bei der Datenhaltung und den Transaktionen eine wichtige Rolle. Dies gilt insbesondere angesichts der großen Verbreitung bspw. von kabellosen Datenübertragungen, *cloud*-basierten Speicherdiensten und Online-Shopping-Plattformen. Zudem gelten bei der Verarbeitung und Speicherung von personengebundenen Daten hohe Anforderungen zum Schutz der Privatsphäre. Daher werden Verfahren der Kryptografie eingesetzt, um u.a. die Vertraulichkeit der Kommunikation in Netzen zu garantieren oder die Identität der beteiligten Kommunikationspartner zweifelsfrei sicherzustellen. Darüber hinaus müssen alle Komponenten einer IT-Infrastruktur gegen vielfältige Bedrohungen und Angriffe abgesichert sein. Man unterscheidet verschiedene Sicherheitsziele, die als Kriterien zur Beurteilung sicherer Informationssysteme dienen. Nachfolgend sollen die wichtigsten Ziele kurz vorgestellt werden.

1.1. Vertraulichkeit

Das klassische Problem beim Austausch von Daten und Nachrichten ist die *Vertraulichkeit* bzw. Geheimhaltung. Der Inhalt von Nachrichten soll nur autorisierten Personen zugänglich gemacht werden. Es stellt sich also die Frage: Wie kann Alice eine Mitteilung über offene Netzwerke an Bob senden, ohne dass Mallory sie durch Überwachung des Datenverkehrs ebenfalls lesen kann? (In der Literatur werden fiktive Personen zum besseren Verständnis der Szenarien eingesetzt. Die beiden eigentlichen Kommunikationspartner werden als Alice und Bob bezeichnet. Mallory ist die Figur des Bösen, die die Kommunikation belauscht). Um Vertraulichkeit sicherzustellen, werden kryptografische Verfahren zur Verschlüsselung von Nachrichten eingesetzt (siehe Kapitel 3).

1.2. Integrität

Unabhängig davon ob eine Nachricht geheim bleiben soll oder nicht, haben normalerweise der Absender und der Empfänger ein großes Interesse daran, dass sie unverändert ankommt. Es stellt sich hier die Frage: Wie kann Bob eine Nachricht von Alice erhalten und sicher sein, dass sie nicht von Mallory oder durch einen technischen Fehler auf dem Kommunikationsweg verändert wurde? Ein weiteres Kriterium ist also die *Integrität* (Unversehrtheit) der ausgetauschten Daten oder von Daten, die auf einem beliebigen Medium gespeichert werden. Um dies sicherzustellen kann der Text mit einer Art elektronischem Fingerabdruck versehen werden (Lit. 12, S. 22).

1.3. Authentizität

Die *Authentifikation* stellt sicher, dass eine Nachricht auch wirklich von dem Absender stammt, der vorgibt, der Absender zu sein. Wie kann Bob sicher sein, dass die Nachricht wirklich von Alice stammt und nicht etwa von Malloy frei erfunden wurde? Dies ist folglich die Frage nach der

Authentizität, der Echtheit der Nachricht. Erst der sichere Beweis, dass eine Person auch wirklich die ist, die sie vorgibt zu sein, führt bspw. beim E-Commerce zu befriedigenden Geschäftsabschlüssen. Verfahren zur Authentifikation können u.a. in der Form von Passwort-basierten Logins, biometrischen Systemen und digitalen Signaturen realisiert werden (s. Kapitel 3.2.2).

1.4. Autorisierung

Ergänzend soll auch garantiert sein, dass ein authentifizierter Nutzer nur auf die Daten zugreifen darf, für die er eine Berechtigung bzw. Autorisation besitzt. Dazu muss eine Form der Kontrolle implementiert sein, die den Zugriff auf jede Ressource verlässlich prüft und reglementiert (Lit. 8, S. 313). Viele Computer-Betriebssysteme kennen zu diesem Zweck Verfahren zur Rechteverwaltung, bspw. durch Zugriffssteuerungslisten (Access Control Lists, ACL), mittels derer detailliert festgelegt werden kann, welcher Nutzer eine Zugriffsberechtigung auf welche Objekte und Dienste hat.

1.5. Verbindlichkeit

Damit der Absender einer Nachricht später nicht leugnen kann, dass diese, zum Beispiel eine Bestellung, tatsächlich von ihm stammt, müssen Verfahren zur Sicherung der *Verbindlichkeit* eingesetzt werden. Es wäre durchaus denkbar, dass Alice (fälschlicherweise) nachträglich behauptet, die Nachricht stamme nicht von ihr (Lit. 12, S. 25). Bob hat somit ein Interesse daran sicherzustellen, dass Alice nicht leugnen kann diese Nachricht so gesendet zu haben. Gleichzeitig sollte Bob auch nicht abstreiten können, dass er die Nachricht empfangen hat. Die Verbindlichkeit stellt somit die Beweisbarkeit des Ursprungs und Empfangs einer Nachricht sicher. Dazu gehört neben der oben beschriebenen Authentifizierung auch die sorgfältige Schlüsselgenerierung und Übermittlung, die gewährleisten, dass keine andere Person geheime Schlüssel kennen kann. Hinzu kommen Zertifikate, die eine vertrauenswürdige Stelle ausgestellt hat und die untrennbar mit der Identität des Besitzers verbunden sind. Erst diese Maßnahmen können einen rechtsverbindlichen Geschäftsabschluss sichern (s. Kapitel 3.2.3).

1.6. Verfügbarkeit

Wenn ein Informationssystem den berechtigten Nutzern jederzeit den Zugriff auf die geforderten Daten oder Dienstleistungen bietet, gilt das Schutzziel der *Verfügbarkeit* als erfüllt. Die Verfügbarkeit kann beispielsweise durch technische Fehler in IT-Komponenten beeinträchtigt werden oder durch andere Nutzer, die das System überlasten. Dies kann in der Form von *Denial-of-Service*-Angriffen geschehen, bei denen durch eine übermäßige Anzahl von sinnlosen Anfragen eine derart große Last auf den angegriffenen Systemen provoziert wird, dass sie nicht mehr in der Lage sind, weitere Anfragen zu beantworten (Lit. 8, S. 310).

1.7. Weitere Schutzziele

Ein weiteres Ziel ist in manchen Situationen die *Anonymität*. So ist es beim Persönlichkeitsschutz notwendig, dass bestimmte Angaben zur eigenen Person nicht publiziert werden. Es kann aber auch notwendig sein, eine Vertraulichkeit nicht nur für den Nachrichteninhalt, sondern sogar während eines vollständigen Kommunikationsvorgangs sicherzustellen. Nur so kann gewährleistet werden, dass übermittelte oder gespeicherte Daten keiner konkreten Person z.B. bei einem verschlüsselten, elektronischen Bezahlvorgang zuordnet werden können. Als abgeschwächte Form der Anonymität kann *Pseudomisierung* gesehen werden, bei der personenbezogene Daten so verändert werden, dass sie ohne Kenntnis des verwendeten Pseudomisierung-Verfahrens nicht mehr einem Individuum zugeordnet werden können (Lit. 5, S. 13).

2. Angriffsszenarien und Gegenmaßnahmen

Die vorgenannten Schutzziele können oftmals durch vorsätzliche Handlungen (Angriffe) oder durch andere Ursachen gefährdet werden. Neben absichtlichen Angriffen durch Cracker zu Zwecken der Datenmanipulation, Spionage oder Sabotage können auch andere Faktoren wie Höhere Gewalt (Naturkatastrophen, Arbeitsniederlegungen), Fahrlässigkeit (mangelhafte Programmierung), technisches Versagen oder organisatorische Mängel zu den Ursachen für die Verletzung von Schutzzielen zählen (Lit. 5, S. 17).

Bei den Angriffen werden zwei wesentliche Typen unterschieden. Mittels *passiven Angriffen* wird versucht, Kommunikationsinhalte auszuspähen oder zu belauschen und somit das Schutzziel der Vertraulichkeit zu verletzen. Da durch sie keine Veränderung der übermittelten oder gespeicherten Daten erfolgt, sind sie schwerer zu detektieren. *Aktive Angriffe* zielen darauf ab, Daten und Nachrichten durch konkrete Eingriffe zu manipulieren, zu verfälschen oder die Kommunikation gänzlich zu unterbinden (Lit. 8, S. 313). Eine Auswahl von weit verbreiteten Angriffsszenarien und Bedrohungen wird nachfolgend kurz vorgestellt. Für Details zu den entsprechenden Angriffen und weiteren Szenarien sei auf die Literatur verwiesen (Lit. 5, Lit. 8).

2.1. Buffer-Overflow

Durch einen Pufferüberlauf-Angriff wird versucht, einen Fehler in der Programmierung eines Software-Produkts auszunutzen. Dabei wird von der Software nicht jede Benutzereingabe korrekt auf die zulässige Länge der Eingabe überprüft. Dadurch kann es dazu kommen, dass bei der Verarbeitung der Daten durch die Software der interne Speicher des Systems in nicht ordnungsgemäßer Weise überschrieben wird („Überlauf“). Auf diesem Weg können Angreifer eigenen Programmcode in das System einschleusen und zur Ausführung bringen. Gegenmaßnahmen bestehen in der konsequenten Kontrolle des Programmcodes bei der Entwicklung, insbesondere im Bereich der Verarbeitung von Benutzereingaben bzw. von Daten, die über Schnittstellen extern eingebunden werden. Zudem sollten Programme möglichst abgeschirmt von anderen Softwareprodukten ausgeführt werden (bspw. in sog. „Sandboxes“), so dass kein Zugriff auf die Speicherbereiche anderer Komponenten besteht.

2.2. Computerviren, Würmer und Trojanische Pferde

Computerviren sind Befehlsfolgen, die zur Ausführung ein weiteres Programm („Wirt“) benötigen. Sie können sich in anderen Softwareprodukten „einnisten“ und somit bei der Ausführung dieses Wirtprogramms ihre Schadwirkung entfalten. Im Gegensatz dazu sind *Computerwürmer* eigenständige Programme, die von einem Nutzer oder einer Systemanwendung aufgerufen werden müssen. Beide Typen sind in der Regel zur Reproduktion fähig (Lit. 5, S. 55ff). Ein *Trojanisches Pferd* ist eine Schadsoftware, die dem Nutzer eine Funktion vortäuscht (und diese auch erfüllt), aber im Hintergrund weitere, versteckte Funktionen durchführt. Sie bildet inzwischen den Großteil der im Internet verbreiteten Schadsoftware (Lit. 5, S. 81).Viren und Würmer verbreiten sich häufig über den Anhang von vermeintlich legitimen E-Mails, die von arglosen Nutzern geöffnet und damit ausgeführt werden (*Social Engineering*). Andere Verbreitungswege insbesondere von Würmern sind ungeschützte Lücken in Netzwerksoftware (bspw. *Lovesan*-Wurm 2003). Weitere Quellen sind zunehmend so genannte *Drive-by-Downloads*, über die eine Schadsoftware bereits beim Aufrufen einer infizierten Website unbemerkt durch die Ausnutzung einer Lücke in der Browser-Anwendung heruntergeladen werden kann. Gegenmaßnahmen gegen diese Typen von Schadsoftware sind die Schulung von Nutzern in Bezug auf mögliche Arten von Social-Engineering-Angriffe und der Einsatz von aktueller Anti-Viren-Software.

2.3. Bot-Netze

Über Bot-Netze können Rechner ferngesteuert werden, die über ein Trojanisches Pferd mit einer Schadsoftware infiziert wurden. Diese *Bots* bzw. *Zombies* werden oftmals für das massenhafte Versenden von Spam genutzt. Ebenfalls sind sie ein wesentlicher Bestandteil für das Durchführen von *Denial-of-Service*-Angriffen, bei denen (Web-)Server durch massenhafte sinnlose Zugriffe überlastet werden sollen (Verletzung des Schutzziels der Verfügbarkeit). Eines der größten Bot-Netze war das russische *BredoLab*, das zwischen 2009 und 2010 bis zu 30 Millionen PCs unter seiner Kontrolle hatte und darüber bis zu 3,6 Milliarden Spam-E-Mails pro Tag versendete.

2.4. DNS-Spoofing, Web-Server-Spoofing

Bei Spoofing-Angriffen wird versucht, den Nutzer unbemerkt auf eine andere Website umzuleiten, ihm dort legitime Inhalte vorzutäuschen und ihn somit zur Preisgabe sensitiver Daten zu verleiten (bspw. Online-Banking-Daten). Dies kann auf IP-Netzwerk-Ebene geschehen indem durch Manipulation der *Domain-Name-System*-Kommunikation (DNS) dem Rechner des Nutzers eine falsche IP-Adresse zur ursprünglich angeforderten WWW-Adresse zurückgegeben wird. Verbreitet sind auch homographische Angriffe, die darauf abzielen, dass der Nutzer in der Schreibweise ähnliche Webadressen aufruft (bspw. Verwendung von „i“ statt „l“ oder Einsatz von kyrillischen Schriftzeichen im Domainnamen) oder den eigenen Tippfehler in der Adresse nicht bemerkt. Abwehrmaßnahmen gegen Angriffe auf DNS-Ebene werden derzeit noch entwickelt (sicheres DNS: DNSSEC). Zur Vermeidung von homographischen Angriffen sollten URLs von sicherheitssensitiven Websites (z.B. Online-Banking-Site) nur aus bekannten und geprüften Lesezeichensammlungen aufgerufen werden.

2.5. Cross-Site-Scripting

Angriffe mittels *Cross-Site-Scripting* (XSS) gehören gegenwärtig zu den am häufigsten auftretenden Sicherheitsrisiken in Web-Anwendungen (Lit. 5, S. 174). Dabei versucht ein Angreifer Schadcode (in der Regel in der Scriptsprache *JavaScript*) über unzureichend gesicherte Formulare oder andere Eingabefunktionen in eine fremde Website einzuschleusen (etwa in ein Gästebuch oder Forum). Sobald ein Nutzer diese infizierte Website aufruft, wird der JavaScript-Code in seinem Browser ausgeführt. Auf diese Weise können *Drive-by-Downloads* realisiert werden (s.o.). Anbieter von Web-Anwendung sollten daher sicherstellen, dass alle Eingaben von Dritten immer auf ihre Zulässigkeit geprüft und ungültige Bestandteile (wie JavaScript-Code) gegebenenfalls herausgefiltert werden.

2.6. SQL-Injection

Auch bei dem Angriffstyp *SQL-Injection* werden Lücken in Web-Anwendungen genutzt, um Schadcode in Websites einzufügen. Bei diesem Typus wird analog zu *Cross-Site-Scripting* versucht, über Eingabefelder oder URL-Parameter speziell formulierte Datenbankabfragen (in der Datenbank-Abfragesprache SQL) in die Web-Anwendung einzuschleusen (injizieren), die dann serverseitig von der Web-Anwendung als Teil eines legitimen Kommandos interpretiert und ausgeführt werden. Dies kann zu unerlaubten Zugriff auf geschützte Datenbereiche führen oder Manipulationsmöglichkeiten von Datenbeständen eröffnen. Zur Vermeidung von SQL-Injection-Angriffen ist es zwingend erforderlich, dass Entwickler von datenbankgestützten Web-Anwendungen Funktionen implementieren, die alle Nutzereingaben vor einer Übergabe an eine Datenbank prüfen und filtern.

2.7. Konstruktionsprinzipien sicherer Systeme

Um sichere Systeme zu entwerfen, sind von Jerome Saltzer und Michael Schroeder (Lit. 9) in den 1970er Jahren einige wesentliche Prinzipien entworfen worden, die bei dem Design- und Entwicklungsprozess (*Security Engineering*) von sicheren Systemen und Programmen berücksichtigt werden sollten.
Zu diesen Prinzipien gehört u.a. das *Erlaubnisprinzip*, das fordert, dass zunächst jeder Zugriff auf Datenbestände oder Programmfunktionen verboten sein soll und erst nach einer ausdrücklichen Erlaubnis gewährt werden darf. Das *Vollständigkeitsprinzip* soll gewährleisten, dass jeder einzelne Zugriff jederzeit auf seine Zulässigkeit geprüft werden muss. Das *Prinzip der Benutzerakzeptanz* besagt, dass Sicherheitsmaßnahmen auf die Kooperation der Nutzer angewiesen sind und daher möglichst verständlich und einfach bedienbar sein sollten. Das *Prinzip der minimalen Rechte* schreibt fest, dass jeder Nutzer nur genau die Rechte erhalten soll, die er zur Erledigung seiner jeweiligen Aufgabe benötigt. Weitergehende Zugriffsrechte dürfen nicht vergeben werden. Schließlich fordert das *Prinzip des offenen Entwurfs*, dass die Sicherheit eines Systems nicht von der Geheimhaltung der eingesetzten Verfahren abhängen darf (*security through obscurity*), sondern alle Funktionsmechanismen offen gelegt werden sollten. Dieses Prinzip ist insbesondere auch bei der Entwicklung von kryptografischen Verfahren von großer Bedeutung (Lit. 5, S. 188), die im folgenden Kapitel beschrieben werden.

3. Kryptografische Verfahren

Kryptografie ist die Lehre der Verschlüsselung von Daten. Kryptografische Verfahren werden genutzt, um einzelne Schutzziele zu realisieren. So wird beim Schutzziel der Vertraulichkeit ein Klartext durch eine Verschlüsselungsmethode in einen Geheimtext bzw. Chiffrat, überführt, hierdurch kann eine fremde Person den ursprünglichen Text nicht mehr erkennen. Der Verschlüsselungsvorgang wird als Chiffrieren, der Entschlüsselungsvorgang als Dechiffrieren bezeichnet (Lit. 12, S. 19).
Genau genommen muss zwischen der Kryptografie, die sich mit der Verschlüsselung von Daten beschäftigt, und der *Kryptoanalyse*, die sich mit derer Entschlüsselung beschäftigt, unterschieden werden. Der Oberbegriff für beide Disziplinen ist *Kryptologie* (Lit. 11, S. 1).

3.1. Verfahren zur Gewährleistung von Vertraulichkeit

Sogenannte *Konzelationssysteme* (Verschlüsselungssysteme bzw. Kryptosysteme) sind Systeme, die zur Geheimhaltung von Daten oder Nachrichten genutzt werden. Der Illusion, dass das verwendete Verschlüsselungsverfahren nicht überwindbar sei, sollte man sich allerdings nicht hingeben. Aus der Tatsache, dass stets die Möglichkeit besteht, dass der verwendete Verschlüsselungsalgorithmus einem Dritten bekannt ist (*Shannons Maxime*), folgt eine weitere Grundregel der Kryptografie, die sogenannte *Kerckhoffs Maxime*. Die Sicherheit eines kryptografischen Verfahrens beruht nicht auf der Geheimhaltung des verwendeten Algorithmus sondern alleine auf der Geheimhaltung des Schlüssels, der zum Dechiffrieren benötigt wird (Lit. 11, S. 8). Aus ihr folgt, dass ohne Kenntnis des Schlüssels kein Rückschluss vom Geheimtext auf den Klartext möglich ist, selbst bei Bekanntsein des verwendeten Verschlüsselungsalgorithmus. Wird ein Geheimtext ohne Kenntnis des Schlüssels entschlüsselt, gilt das Verschlüsselungsverfahren als *gebrochen* bzw. umgangssprachlich als *geknackt*. Ein Schlüssel gilt als *kompromittiert*, wenn er bspw. durch Diebstahl entwendet wurde (Lit. 8, S. 316).

3.1.1. Symmetrische Verschlüsselung

Symmetrische Verschlüsselungsverfahren werden bereits seit vielen Jahrhunderten genutzt. Sender und Empfänger haben sich dabei entweder auf einen Schlüssel geeinigt (*Secret Key*) oder der

Dechiffrierschlüssel lässt sich aus dem Chiffrierschlüssel berechnen und umgekehrt (Lit. 1, S. 154). Symmetrische Verschlüsselungsverfahren werden auch als *Secret-Key-Verfahren* bezeichnet. Dabei werden in der Regel zwei grundlegende Verfahren unterschieden. Bei *Substitutionsverfahren* werden die Zeichen einer Nachricht durch andere Zeichen ersetzt. *Transpositionsverfahren* hingegen vertauschen die Positionen der Zeichen einer Nachricht. Diese beiden Verfahren werden oftmals in mehreren Durchgängen (Runden) angewendet und dabei miteinander kombiniert (Lit. 8, S. 317).

3.1.1.1. Historische Verfahren

Die berühmteste Verschlüsselung ist die von Gaius Julius Caesar. Er nutzte ein Substitutionsverfahren, um jeden der 20 Buchstaben des lateinischen Alphabets um drei Stellen nach rechts zu verschieben. Da diese zyklische Vertauschung mathematisch wie eine Addition von 3 (mit den Sonderregeln 18+3=1, 19+3=2, 20+3=3) funktioniert, nennt man das Verfahren auch *Caesar-Addition*. Die heute älteste bekannte Verschlüsselung ist die *Skytale* von Sparta (5. Jhdt. v. Chr.), die auf einem einfachen Transpositionsverfahren basiert. Ein Holzstab wurde mit einem schmalen Band aus Pergament spiralförmig umwickelt und dann der Länge nach mit einer Nachricht beschrieben. Den Text auf dem abgewickelten Pergamentstreifen sollten nur die Generäle lesen können, die über Stäbe vom gleichen Durchmesser verfügten (Lit. 2, S. 3). Im 16. Jhdt. entwickelte Blaise de Vigenère die Caesar-Methode weiter, indem er den Verschiebungsbetrag fortlaufend änderte, es wird somit eine Folge von Zahlen als Schlüssel, z.B. 12, 1, 19, 6, 2 auf den Klartext angewendet. Der erste Buchstabe wird um 12 Zeichen, der zweite um 1 Zeichen usw. verschoben. Nach dem Ende der Folge wird wieder von vorne angefangen. Man könnte sich den Schlüssel auch als Sequenz von Buchstaben (NBUGC; A wird um 12 Stellen auf N verschoben, usw.) vorstellen. Das *Vigenère*-Verfahren machen sich die sogenannten Rotormaschinen wie die im zweiten Weltkrieg verwendete Chiffriermaschine *Enigma* zu Eigen. Sie bestehen aus mehreren hintereinander liegenden drehbaren Scheiben. Jede dieser Scheiben weist vorne und hinten 26 Kontaktflächen auf, für jeden Buchstaben des Alphabets eine. Die Scheiben sind fest verdrahtet. Dies führt zu einer Permutation des Alphabets und somit zu einer einfachen Ersetzung von Buchstaben. Bereits bei drei Rotoren sind Schlüsselwörter mit einer Periodenlänge von 26^3 = 17.576 Zeichen möglich. Der Schlüssel besteht in der Angabe, welcher Rotor in welcher Reihenfolge einzusetzen ist und wie ihre jeweilige Anfangsstellung aussieht. Die Enigma bestand aus einer Kombination von bis zu acht austauschbaren Rotoren, die nach jedem Zeichen jeweils um einen anderen Betrag weitergeschaltet wurden. Zusätzlich besaß sie einen Reflektor, der dafür sorgte, dass jedes Zeichen zweimal in unterschiedlicher Richtung das Gerät durchlief, hinzu kam ein weiterer paarweiser Austausch von Zeichen, der je einmal am Anfang und am Ende der Operation durchgeführt wurde. Zum Schlüssel gehörte hier auch die Angabe, wie die Zeichenersetzung vorzunehmen war.
Ist der Schlüssel genauso lang wie der zu chiffrierende Text, handelt es sich um das *One-Time-Pad* (Einmalblock)-Verfahren. Dies ist auch das einzige Verfahren, dessen Sicherheit mathematisch bewiesen wurde. Natürlich muss bei diesem Verfahren jedes Mal ein neuer Schlüssel verwendet werden (Lit. 2, S. 51).

Die meisten aktuellen Verschlüsselungsverfahren arbeiten mit einem weiteren Trick. In jedem Verschlüsselungsschritt werden nicht Zeichen für Zeichen, sondern ein längerer Klartextblock verarbeitet und durch den Geheimtextblock ersetzt, wobei jedes Klartextzeichen eines Blocks das gesamte Ergebnis beeinflusst (Lit. 12, S. 42). Dadurch werden Regelmäßigkeiten im Klartext über mehrere Zeichen hinweg verteilt (*Diffusion*). Ein Chiffrierungsschritt muss dabei so beschaffen sein, dass zwei Klartextblöcke, die sich nur in einem Zeichen unterscheiden, zu völlig unterschiedlichen Geheimtextblöcken führen. Diese Verfahren werden Blockverschlüsselungen genannt. Heutzutage werden Methoden mit mindestens 8 Byte, also 64 Bit, verwendet. Bei vielen Verfahren wird der Eingabeblock zu Anfang einer jeden Runde in zwei Hälften (R und L) zerlegt.

Die Operationen (Kombination von Addition, Multiplikation, exklusives oder (XOR) und Vertauschungen) werden nur auf den R-Teil angewendet, ihr Ergebnis wird durch XOR mit L verknüpft und bildet die rechte Hälfte des Rundenergebnisses, während die linke Hälfte von dem unveränderten R gebildet wird. So wird für eine Hälfte des Blocks mit der anderen Hälfte *Konfusion* erzeugt, während die andere Hälfte unverändert bleibt. Diese wird in der jeweils folgenden Runde dem Konfusionsverfahren unterworfen. In jede einzelne Runde gehen Teile des Schlüssels ein. Verfahren mit dieser Methode werden als *Feistel-Netzwerke* bezeichnet. Die Dechiffrierung gestaltet sich bei Kenntnis des Schlüssels einfach, denn sie läuft in umgekehrter Reihenfolge die Inversen der elementaren Operationen durch und liefert als Ergebnis den Klartext.

3.1.1.2. DES und seine Varianten

Das bekannteste und lange Zeit am weitesten verbreitete symmetrische Verschlüsselungsverfahren ist der *Data Encryption Standard* (DES). Das Verfahren wurde 1976 in den Vereinigten Staaten als Bundesstandard anerkannt, es verwendet eine Blocklänge von 64 Bit sowie eine Schlüssellänge von 56 Bit und wird 16mal durchlaufen. Dadurch ergibt sich ein Schlüsselraum von 2^{56} unterschiedlichen Schlüsseln. DES ist auf Standardrechnern jedoch durch Ausprobieren aller möglichen Schlüssel (*Brute-Force-Attack*) in weniger als 24 Stunden zu brechen, daher gilt DES als technisch veraltet. Eine auch heute noch sichere Variante von DES ist *Triple-DES*, die dreimalige, hintereinander geschaltete Anwendung von DES. Die Schlüssellänge steigt dadurch auf 168 Bit (etwa $3{,}74 \times 10^{50}$ mögliche Schlüssel).

3.1.1.3. AES

Bei dem *Advanced Encryption Standard* (AES) handelt es sich um den Nachfolger von DES. Bei AES (ursprünglich als *Rijndael*-Algorithmus entwickelt) handelt es sich um eine frei verfügbare symmetrische 128-Bit-Blockchiffre mit Schlüssellängen von 128, 192 und 256 Bit, die schneller als Triple-DES arbeitet. Sie ist seit 2001 in den USA standardisiert und wird auch vom amerikanischen Geheimdienst für Daten der höchsten Geheimhaltungsstufe verwendet (Lit. 10, S.119). Bei einer AES-Schlüssellänge von 256 Bit gelten Brute-Force-Attacken mit heutiger Computerleistung als nicht realistisch durchführbar, es ist jedoch nicht ausgeschlossen, dass sich in Zukunft durch die wachsende Hardwareleistung neue Angriffspunkte anbieten werden.

3.1.1.4. AES-Alternativen

Einen guten Kompromiss zwischen Sicherheit und Verfügbarkeit stellt der *Blowfish*-Algorithmus von Bruce Schneier dar. Die Blocklänge beträgt 64 Bit, die Schlüssellänge kann bis zu 448 Bit beliebig gewählt werden und der Algorithmus wird 16mal durchlaufen. Der Nachfolge-Algorithmus *Twofish* verschlüsselt 128-Bit-Blöcke und nutzt eine Schlüssellänge von bis zu 256 Bit.

Als weitere Alternative bei der Suche nach dem Nachfolger für DES war der Algorithmus *Serpent* ein vielversprechender Kandidat. Er verwendet ebenfalls eine 128-Bit-Blocklänge, gilt allerdings als vergleichsweise zukunftssicher durch besonders große Sicherheitspuffer im Algorithmus, die auch bei einer markanten Leistungssteigerung gängiger Hardware kaum Angriffspotentiale erwarten lassen. Er ist dadurch jedoch auch deutlich langsamer als die Konkurrenten *Rijndael* und *Twofish*.

Der in Japan entwickelte Algorithmus *KASUMI* wird vor allem im Mobilfunkbereich verwendet (im UMTS-Standard). Er ähnelt dem DES-Algorithmus und ist sehr effizient und schnell. Die Schlüssellänge und die Blocklänge betragen 128 Bit (Lit. 10, S. 132).

3.1.2. Asymmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung besteht immer die Notwendigkeit, den zu verwendenden Schlüssel über einen sicheren Kanal auszutauschen. Falls dieser Kanal nicht sicher ist, kann der Schlüssel abgehört werden. Mitte der 1970er Jahre veröffentlichten Whitfield Diffie und Martin Hellman sowie unabhängig von ihnen Ralph Merkle ein Verfahren, das dieses Problem des Schlüsselaustauschs löst, indem zum Chiffrieren ein anderer Schlüssel als zum Dechiffrieren verwendet wird (asymmetrisches Verfahren). Zusätzlich darf es nicht möglich sein aus der Kenntnis eines Schlüssels den jeweils anderen abzuleiten. Wer ein solches Verfahren nutzt, muss zunächst ein Paar zusammengehörender Schlüssel generieren. Einen der beiden Schlüssel hält er geheim (*Private Key*), den anderen gibt er der Öffentlichkeit bekannt (*Public Key*). Diese Verfahren werden daher auch *Public-Key-Verfahren* genannt. Jeder, der nun eine verschlüsselte Nachricht an eine Person schicken will, besorgt sich deren frei verfügbaren öffentlichen Schlüssel, verschlüsselt seine Nachricht damit und verschickt den Geheimtext. Dieser so chiffrierte Text kann nur vom Empfänger mit seinem privaten, geheimen Schlüssel dechiffriert werden (Lit. 2, S. 94). Von zentraler Bedeutung ist somit, dass der Empfänger einer Nachricht den verwendeten Schlüssel vorgibt, nicht der Sender.

3.1.2.1. RSA

RSA, benannt nach den Entwicklern Ronald L. Rivest, Adi Shamir und Leonard M. Adleman, ist das bekannteste Public-Key-Verfahren und ein Quasi-Standard im Internet. Das Prinzip beruht darauf, dass es kein Problem darstellt, zwei große Primzahlen miteinander zu multiplizieren, es aber praktisch unmöglich ist, aus dem Produkt wieder die beiden Faktoren zu ermitteln. Dabei ist zu beachten, dass die beiden Faktoren sich in ihrer Länge deutlich unterscheiden. In praktischen Anwendungen variiert das Produkt zwischen 512 Bits (geringe Sicherheit) und 2048 Bits (sehr hohe Sicherheit) (Lit. 1, S.207). Es wird allgemein angenommen, dass der Aufwand zur Wiederherstellung des Klartextes aus dem Chiffretext und dem öffentlichen Schlüssel äquivalent zur Faktorisierung des Produktes der beiden Primzahlen ist, allerdings gibt es dafür noch keinen mathematischen Beweis. RSA ist um den Faktor 1000 langsamer als AES. Dies mag als ein Nachteil von RSA erscheinen, ist aber tatsächlich eher von Vorteil. Denn für die Ver- und Entschlüsselung von normalen Mitteilungen fällt diese Zeit praktisch nicht ins Gewicht. Wer aber RSA mittels einer Brute-Force-Attacke brechen möchte, tut sich umso schwerer, je langsamer der Algorithmus ist. Auch bei alternativen Angriffsszenarien (bspw. Faktorisierungsangriff) gibt es derzeit keine realistische Perspektive, dass RSA bei ausreichender Schlüssellänge (2048 Bits und mehr) überwunden werden könnte.

3.1.2.2. ElGamal

Das Prinzip des 1985 von Taher ElGamal entwickelten Algorithmus beruht auf dem Problem des "diskreten Logarithmus“ (Lit. 3, S. 127). In praktischen Anwendungen variiert die Schlüssellänge zwischen 512 Bits (geringe Sicherheit) und 1024 Bits (sehr hohe Sicherheit). Eine Variante des ElGamal-Verfahrens ist der 1991 entwickelte Digital Signature Algorithm (DSA), der 1994 von der US-Standardisierungsbehörde NIST zum Digital Signature Standard (DSS) erklärt wurde und ebenso wie RSA für digitale Signaturen verwendet werden kann (Lit. 11, S. 555).

3.1.3. Hybride Verschlüsselung

Da asymmetrische Verschlüsselungssysteme in der Regel sehr viel langsamer arbeiten als symmetrische Algorithmen, werden bei den im Internet gebräuchlichen Verschlüsselungsprogrammen häufig beide Verfahren eingesetzt. Bei einem Verbindungsaufbau erzeugt der Sender einen zufälligen Sitzungsschlüssel (*Session Key*), mit dem er die Nachricht

verschlüsselt. Der Session Key wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und zusammen mit der verschlüsselten Nachricht verschickt. Der Empfänger kann dann mit seinem privaten Schlüssel den asymmetrisch chiffrierten Schlüssel dechiffrieren und mit ihm den die symmetrisch chiffrierte Nachricht dechiffrieren. Durch diese Kombination (hybride Verschlüsselung) vereinigt man einen gesicherten, aber langsamen Schlüsseltausch mit einer schnellen, aber weniger sicheren Verschlüsselung.

3.2. Verfahren zur Gewährleistung der Integrität und der Authentizität

Moderne kryptografische Verfahren lassen sich aber nicht nur einsetzen um Vertraulichkeit, sondern auch die drei Ziele Integrität, Authentizität und Verbindlichkeit zu erreichen. Man spricht dabei von *Authentifikationssystemen*. Dazu sind sogenannte *Hashfunktionen* erforderlich. Hashfunktionen sind mathematische Methoden, die aus einem beliebigen Klartext nach einem bestimmten Verfahren einen Fingerabdruck (Prüfsumme, Hashwert, Message Digest (MD)) der Nachricht generieren. Die Funktion verwandelt einen Klartext so in einen MD, dass auch die kleinste Veränderung des ursprünglichen Texts zu einem gänzlich anderen MD führt. Somit kann überprüft werden, ob der Text verändert wurde. Hashfunktionen sind nicht umkehrbar und gelten somit als Einwegfunktionen. Anders als beim Chiffrieren darf eine Wiederherstellung des Klartextes aus dem einmal erzeugten MD nicht möglich sein. Es kann jedoch vorkommen, dass zwei unterschiedliche Klartexte denselben Hash-Wert produzieren (*Kollision*).
Der Vorteil dieses Verfahrens liegt in der Tatsache, dass anstatt des gesamten Textes lediglich ein kurzer MD besonders geschützt werden muss. Die zurzeit bekanntesten Hashfunktionen sind u.a. SHA-1 (Secure Hash Algorithm One), er wurde von der NSA (National Security Agency) entwickelt und als US-Standard angenommen. Der Hashwert hat eine Länge von 160 Bit. RIPEMD (RIPE-Message Digest) wurde im Rahmen des EU-Projektes RIPE (RACE Integrity Primitives Evaluation, 1988-1992) von Dobbertin, Bosselaers und Preneel entwickelt. Generell bieten Hashfunktionen mit längeren Prüfziffern höhere Sicherheit. Gegenwärtig gelten die Nachfolge-Standards SHA-256 und SHA-512 mit jeweils 256-Bit-Hash bzw. 512-Bit-Hash als sicher. Mittels Kollisionsangriffen auf SHA-1 wurden mit leistungsfähiger Computerhardware bereits erste Erfolge erzielt, weshalb dieses Verfahren als technisch veraltet gilt. Der ebenfalls veraltete, aber noch weit verbreitete Standard MD5, von Rivest entwickelt, erzeugt einen MD von 128 Bit Länge und kann mittlerweile auf Standard-PC-Hardware innerhalb weniger Stunden durch Kollisionsangriffe überwunden werden.

3.2.1. Symmetrische Authentifikationssysteme

Der Sender verschlüsselt den MD einer Nachricht und sie selbst mit dem geheimen Schlüssel, der beiden Kommunikationspartnern bekannt ist (Secret Key) und sendet Nachricht sowie MD an den Empfänger. Der Empfänger dechiffriert den MD und die Nachricht. Durch erneute Anwendung der Hashfunktion auf die Nachricht und Vergleich des Ergebnisses mit dem entschlüsselten MD kann er feststellen, ob die Nachricht während der Übertragung verändert wurde (Lit. 2, S. 69). Ein solches System hat einige Nachteile. Zum einen kann nur eine Person, die den geheimen Schlüssel kennt, eine solche Überprüfung vornehmen; wünschenswert wäre aber in vielen Situationen, dass jeder Beliebige die Echtheit einer Nachricht überprüfen kann. Zum anderen kann jeder, der über den zur Überprüfung nötigen Schlüssel verfügt, auch authentifizierte Nachrichten erstellen. Das bedeutet, dass das System in Gruppen von mehr als zwei Teilnehmern dem Empfänger keine Auskunft mehr darüber gibt, von wem eine bestimmte Nachricht eigentlich stammt, und dass es auch bei nur zwei Teilnehmern stets möglich ist, das Erstellen einer bestimmten Nachricht abzustreiten. Den MD könnte genauso gut der jeweils andere Kommunikationspartner verschlüsselt haben, denn auch er hat den Schlüssel (Lit. 7, S. 108). Integrität und Authentizität einer Nachricht werden also nur gegen Angriffe von außen stehenden Personen gesichert, Verbindlichkeit dagegen wird überhaupt nicht erreicht, da alle beteiligten Kommunikationspartner den geheimen Schlüssel besitzen.

3.2.2. Asymmetrische Authentifikationssysteme und digitale Signaturen

Erst die Kombination aus asymmetrischer Verschlüsselung und Hashwert bietet die Möglichkeit, ein Analogon zur menschlichen Unterschrift zu erzeugen, in diesem Zusammenhang wird von *digitalen Signaturen* gesprochen (Lit. 2, S. 115). Will jemand eine Nachricht als von ihm erstellt ausweisen (quasi unterzeichnen), wendet er eine Hashfunktion auf diese Nachricht an, den MD verschlüsselt er mit seinem Private Key und hängt das Ergebnis als digitale Signatur der zu übertragenden unverschlüsselten Nachricht an. Jeder, der im Besitz des zugehörigen Public Key ist, kann die Echtheit der Nachricht überprüfen indem er den MD dechiffriert und diesen mit dem von ihm neu berechneten MD vergleicht, der sich aus der unverschlüsselten Nachricht ergibt (gleiche Hashfunktion verwenden). Sind diese Werte identisch, wurde die Nachricht unterwegs nicht verändert. Signieren kann die Nachricht nur der Besitzer des Private Keys, so dass Integrität, Authentizität und Verbindlichkeit realisiert werden können. Allerdings kann bei diesem Verfahren jeder den Klartext lesen, da nur der MD verschlüsselt wurde.
Will man auch noch Vertraulichkeit sicherstellen, muss das Verfahren erweitert werden (*elektronischer Umschlag*). Der mit dem Private Key des Senders chiffrierte MD wird der Nachricht angehängt. Die auf diese Weise verlängerte Nachricht wird nun mit dem Public Key des Empfängers chiffriert und übermittelt. Der Empfänger dechiffriert die verlängerte Nachricht mit seinem Private Key und trennt den immer noch chiffrierten MD von der nun dechiffrierten Nachricht ab, den er mit dem Public Key des Senders entschlüsselt. Er berechnet selbst den MD der Nachricht und vergleicht diesen mit dem vom Sender übermittelten MD. Stimmen sie überein, kann er sicher sein, dass die Nachricht vom Sender stammt, unterwegs nicht verändert und von keinem Dritten belauscht wurde (Lit. 12, S. 114).

3.2.3. Zertifizierungsinstanzen und Public-Key-Infrastrukturen

Eine vollständige Sicherstellung von Verbindlichkeit und Authentizität kann aber auch durch diese asymmetrischen Kryptografie-Verfahren allein noch nicht garantiert werden. Es besteht zunächst noch kein nachvollziehbarer Zusammenhang zwischen einem Private Key und der vorgeblich zu ihm gehörenden Person. Jemand kann sich als eine andere Person ausgeben, indem er unter deren Namen einen selbst erzeugten Private Key in Umlauf bringt. Dieses Problem kann durch das Einschalten eines vertrauenswürdigen Dritten gelöst werden, der sich für die Identität einer Person verbürgt. Dies kann über Vertrauensnetzwerke (Web of Trust) oder offizielle Zertifizierungsinstanzen (Certification Authority (CA), Trustcenter (TC)) geschehen. CAs liefern mit digitalen Zertifikaten und Schlüsseln die Grundausstattung zur Teilnahme am rechtsverbindlichen und vertraulichen elektronischen Geschäftsverkehr. Sie überprüfen zunächst die Identität des Nutzers und generieren einen elektronischen Ausweis (Zertifikat), das bestätigt, dass der Public Key wirklich der beantragenden Person gehört (Lit. 3, S. 208). An dieses CA kann sich der Empfänger wenden und den Public Key des Senders abrufen. Von besonderer Bedeutung ist hierbei, dass die jeweiligen CAs nicht kompromittiert werden dürfen, da dies zu einem Zusammenbruch des Vertrauensnetzwerks führen kann. Diese Vertrauensnetzwerke werden auch als Public-Key-Infrastukturen bezeichnet. Das Format solcher Zertifikate lässt sich standardisieren, so dass sie automatisch auswertbar sind. Beispielhafte Standards zum Austausch derartiger Zertifikate sind X.509, PKIX und OpenPGP.

Die Rechtsgültigkeit entsprechender digitaler Signaturen und Zertifikate und die Anforderungen an diese sind in der internationalen Gesetzgebung festgeschrieben. Das entsprechende deutsche *Signaturgesetz* (SigG) in der Fassung von 2001 lehnt sich in den meisten Aspekten an die EG-Signaturrichtlinie 1999/93/EG an. Ergänzt wird es durch die Einzelregelungen in der *Signaturverordnung* (SigV). Ziel ist es, allgemein anerkannte und abgesicherte Rahmenbedingungen zur Nutzung einer „elektronischen Unterschrift“ im alltäglichen Geschäftsverkehr zu schaffen.

Es werden dazu in der europäischen Rechtsprechung meist drei Abstufungen von digitalen bzw. elektronischen Signaturen unterschieden:
Die *einfache elektronische Signatur* soll sich als beliebiger Text einer Person zuordnen lassen und stellt keine besonderen Ansprüche an Form und Ausgestaltung dieser Signatur. Die *fortgeschrittene elektronische Signatur* ist eine übliche digitale Signatur, die mit (kryptografischen) Verfahren erstellt wurde und bei der sichergestellt ist, dass sie nur von dem Signaturschlüssel-Inhaber erzeugt werden kann. Die *qualifizierte elektronische Signatur* ergänzt die fortgeschrittene elektronische Signatur durch ein qualifiziertes Zertifikat einer sichereren Signaturerstellungseinheit. Ein derartiges qualifiziertes Zertifikat darf nur durch Anbieter ausgegeben werden, deren Konformität mit dem Signaturgesetz und der Signaturverordnung überprüft wurden (Lit. 10, S. 532).

4. Sicherheit von elektronischen Zahlungssystemen

Kryptografische Verfahren sind von zentraler Bedeutung für die sichere Funktionsweise von elektronischen Zahlungssystemen. Zur Systematisierung von derartigen Zahlungssystemen lassen sich mehrere Klassifikationsverfahren verwenden. So kann nach Art des Zahlungszeitpunkts (Pay before, Pay now, Pay later), der Höhe der Zahlung (Nano-, Micro-, Macro-Payment), des Transaktionswegs (absenderinitiiert, empfängerinitiiert), der Hard/Software-Komponenten oder nach Art der Basierung unterschieden werden. Für Details sei auf Lit. 4 verwiesen.

4.1. Sicherheits-Anforderungen an elektronische Zahlungssysteme

Die Anforderungen Vertraulichkeit, Authentizität, Integrität und Verbindlichkeit müssen insbesondere bei elektronischen Finanztransaktionen ausreichend erfüllt sein. Die Transaktionsdaten müssen vor unbefugtem Zugriff geschützt sein (Vertraulichkeit). Kreditkartendaten, persönliche Angaben von Kunden sowie Details zu den gekauften Waren dürfen nicht an Dritte gelangen. Diese müssen bei allen Transaktionspartnern sicher gespeichert sein, soweit eine Speicherung von Kreditkartennummern bei einem Händler in Kundendatenbanken überhaupt sinnvoll und erforderlich ist. Zudem muss sichergestellt werden, dass der Datentransfer der Zahlungsdetails über sichere Kanäle erfolgt, z.B. durch die Nutzung von verschlüsselten Datenverbindungen (SSL) für Online-Shopping-Angebote. Weiter ist es erforderlich, dass die Teilnehmer an einer finanziellen Transaktion mit Hilfe von digitalen Zertifikaten und Signaturen eindeutig identifiziert werden (Authentizität). In der Praxis überwiegt derzeit die Authentifikation mittels Passwort oder TAN/PIN-Verfahren. Auch die Integrität der übermittelten Transaktionsdaten ist von hoher Bedeutung. Keinesfalls darf es möglich sein, dass einzelne Angaben der Transaktion manipuliert werden können (Zahlbetrag, Empfänger des Geldbetrags, Art der Ware). Ferner ist die Verbindlichkeit beim Zahlungsverkehr wichtig. Keiner der Beteiligten darf abstreiten können, dass er die Zahlung in Auftrag gegeben oder erhalten hat.

Auch Anonymität kann eine weitere Anforderung an elektronische Zahlungssysteme sein. Während übliches Papier-Geld in der Regel anonym im stationären Einzelhandel zum Erwerb von Artikeln oder Dienstleistungen eingesetzt wird, werden elektronische Transaktionen meist aufgezeichnet und sind damit nachvollziehbar. Der Wunsch nach Anonymität beim Zahlungsverkehr steht jedoch im Widerspruch zu den Anforderungen der Verbindlichkeit sowie Authentizität. Um dieses Problem zu lösen, wurden Zahlungssysteme geschaffen, die mithilfe kryptografischer Verfahren einen anonymisierten elektronischen Zahlungsverkehr ermöglichen sollen. Ein Vertreter ist die in Deutschland verwendete *Geldkarte.* Die virtuelle *Bitcoin*-Währung realisiert Anonymität, Verbindlichkeit und Authentizität mittels komplexer Kryptografie-Verfahren und einem anonymen Peer-to-Peer-Netzwerk.

4.2. Zahlungssysteme im World Wide Web

Die größte Bedeutung im Web hat das klassische Zahlungsverfahren *Vorkasse*, bei dem der Kunde mittels Online-Banking den Rechnungsbetrag auf das Bankkonto des Händlers überweist. Auch Zahlung auf Rechnung, per Kreditkarte, Lastschrift und Nachnahme sowie Variationen klassischer Zahlungsverfahren wie Überweisungen via Dienstleister wie *sofortüberweisung.de* sind nach wie vor die bestimmenden Zahlungsverfahren. Bezogen auf den Umsatz bei deutschen Händlern spielt auch *PayPal* zunehmend eine bedeutende Rolle (Lit. 6). Dort kann der Kunde ein virtuelles Konto eröffnen, das er über klassische Zahlungsverfahren wie Kreditkarte oder Überweisung aufladen und das Guthaben dann zum Zahlen auf anderen Plattformen einsetzen kann. Weitere Dienstleister wie *Amazon Payments, Moneybookers/Skrill, Google Checkout* oder *ClickandBuy* setzen auf ähnliche Verfahren, indem sie die Rechnungssumme nach Abzug einer Provision an den jeweiligen Händler weiterreichen. Neue E-Geld-Konzepte wie die vorgenannte *Bitcoin*-Währung verzeichnen zwar ein deutliches Wachstum, spielen jedoch noch eine untergeordnete Rolle im Zahlungsverkehr. Zunehmend verbreiten sich auch zielgruppen-spezialisierte Zahlungsplattformen, wie zum Zwecke von *social microdonations*, hier kann man mit geringem Aufwand Kleinstbeträge in Form von Spenden an Ersteller beliebiger Web-Inhalte übermitteln (bspw. *flattr.com*).

4.3. Mobile Zahlungssysteme

Mobile Bezahlverfahren gewinnen angesichts der starken Verbreitung von Mobiltelefonen zunehmend an Bedeutung. Vor allem auf dem afrikanischen Kontinent haben sich Mobilfunk-basierte Zahlungssysteme für Kleinstbeträge etabliert. Ausgehend von Zahlungsverfahren, die über die Telekommunikationsrechnung abgerechnet werden (kostenpflichtige SMS oder sogenannte Mehrwert-Rufnummern) sind inzwischen auch neue Systeme u.a. in den USA auf den Markt gekommen, die jedem Nutzer mit einem Smartphone die Annahme von Zahlungen via Kreditkarte ermöglichen (*squareup.com*). Auch hier tritt ein Anbieter auf, der eine Plattform zur Verfügung stellt, über den Händler und Kunden Zahlungen abwickeln können. Zunehmend bieten auch klassische Banken in Modellprojekten mobile Zahlungssysteme an (*Popmoney* der Citibank USA). Weitere neue technische Systeme bauen auf der Nahfunktechnik NFC (Near Field Communication) auf, die es Smartphones ermöglicht Zahlungen von Kleinstbeträgen auch ohne die Angabe einer PIN durchzuführen. Das Mobiltelefon soll zu einer Art „mobiler Geldbeutel“ ausgebaut werden. Noch bestehen allerdings offene Fragen hinsichtlich der Sicherheit derartiger Transaktionsverfahren.

Literatur

1 Bauer, Friedrich L.: Entzifferte Geheimnisse. Methoden und Maximen der Kryptologie. 3., überarb. Aufl. Berlin, Heidelberg, New York: Springer-Verlag, 2000, 503 S.

2 Beutelspacher, Albrecht: Kryptologie. Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen; ohne alle Geheimniskrämerei, aber nicht ohne hinterlistigen Schalk, dargestellt zum Nutzen und Ergötzen des allgemeinen Publikums. 6., überarb. Aufl. Braunschweig; Wiesbaden: Vieweg, 2002, 152 S.

3 Buchmann, Johannes: Einführung in die Kryptographie. Berlin, Heidelberg: Springer, 1999, 229 S.

4 Dannenberg, Marius; Ulrich, Anja: E-Payment und E-Billing. Wiesbaden: Gabler, 2004, 272 S.

5 Eckert, Claudia: IT-Sicherheit. Konzepte - Verfahren - Protokolle. 7., überarb. und erw. Auflage. München: Oldenbourg Verlag, 2012, 1004 S.

6 E-Commerce-Center Handel: Internet-Zahlungsverfahren aus Sicht der Händler: Ergebnisse der Umfrage IZH6. Kurzauswertung. Februar 2012.

7 Grimm, Rüdiger: Kryptoverfahren und Zertifizierungsinstanzen. In: Datenschutz und Datensicherheit (DuD), 1996, S. 27 – 36

8 Meinel, Christoph; Sack, Harald: Digitale Kommunikation. Berlin, Heidelberg: Springer, X.media.press, 2009, 422 S.

9 Saltzer, Jerome H; Schroeder, Michael D.: The protection of information in computer systems. Proceedings of the IEEE 63(9): 1278-1308 (1975), http://web.mit.edu/Saltzer/www/publications/protection/ (zuletzt 21.10.2012)

10 Schmeh, Klaus: Kryptografie. Verfahren, Protokolle, Infrastrukturen. 3., überarb. und erw. Auflage. Heidelberg: dpunkt.verlag, 2007, 772 S.

11 Schneier, Bruce: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C. München: Pearson Studium, 2006, 844 S.

12 Selke, Gisbert: Kryptographie. Verfahren, Ziele Einsatzmöglichkeiten. 1. Aufl. Köln: O'Reilly, 2000, 225 S.