Sicherheitsanforderungen und Sicherheitsmechanismen bei IT-Systemen*

Petra Wohlmacher

Universität Klagenfurt
Institute für Informatik – Systemsicherheit
Villacher Str. 161
A-9020 Klagenfurt

petra.wohlmacher@uni-klu.ac.at

ZUSAMMENFASSUNG

Der folgende Beitrag beschreibt die wichtigsten Sicherheitsanforderungen, die heutige IT-Systeme erfüllen müssen. Die dargestellten Sicherheitsmechanismen, die zur Erfüllung der Anforderungen eingesetzt werden, basieren auf modernen kryptographischen Mechanismen sowie auf Sicherheitsinfrastrukturen.

SCHLÜSSELWÖRTER

Sicherheitsanforderung, Sicherheitsmechanismus, Vertraulichkeit, Integrität, Authentizität, Nichtabstreitbarkeit, Kryptosystem, Session-Key-Verfahren, One-Way-Hashfunktion, Trapdoor-One-Way-Funktion, Message Authentication Code, Digitale Signatur, Authentifizierungsprotokoll, Challenge&Response-Protokoll, Sicherheitsinfrastruktur, Trust Center, Public-Key-Infrastruktur, Originalität.

1 Einführung

Informationstechnische[1] Systeme (IT-Systeme) spielen in praktisch allen Bereichen der heutigen Informationsgesellschaft eine essentielle Rolle. Mit den immer höher werdenden Anforderungen an die Leistungsfähigkeit und die Möglichkeiten der IT-Systeme steigen auch die Forderungen nach Sicherheit und Vertrauenswürdigkeit. Besondere Bedeutung haben diese Forderungen für sicherheitsrelevante Anwendungen sowie für Anwendungen, in denen personenbezogene Daten verarbeitet werden.

* Dieser Beitrag wurde akquiriert von Roland Kaschek.

[1] Vgl. [22]: Die Informationstechnik beinhaltet Geräte und Systeme, die auf Elementen, Erkenntnissen und Ergebnissen von Technischer, Praktischer und Angewandter Informatik beruhen, und umfaßt die Gesamtheit der Arbeits-, Entwicklungs-, Produktions- und Implementierungsverfahren in der Computertechnik. Im Vergleich zu einem IT-System ist ein Informationssystem ein System zur Informationsproduktion und Kommunikation für die Deckung von Informationsnachfrage und zur Speicherung, Wiedergewinnung, Verknüpfung und Auswertung von Informationen und besteht aus einer Datenverarbeitungsanlage, einem Datenbanksystem und den Auswertungsprogrammen.

2 Anforderungen, Mechanismen und Bedrohungen

An heutige IT-Systeme werden im wesentlichen die folgenden Sicherheitsanforderungen gestellt, die mit den angegebenen kryptographischen Sicherheitsmechanismen erfüllt werden können:

- Vertraulichkeit (Confidentiality): Informationen sollen nur den dazu berechtigten Parteien (dies können Personen oder auch Geräte sein) zur Verfügung stehen. Um die Informationen gegenüber Unbefugten geheimzuhalten, können Verschlüsselungsverfahren eingesetzt werden.
- Integrität von Daten (Data Integrity): Es soll sichergestellt werden, daß Daten nicht unautorisiert geändert wurden. Unbefugte Änderungen an den Daten können mittels One-Way-Hashfunktionen, Message Authentication Codes und digitaler Signaturen erkannt werden.
- Verfügbarkeit (Availability): Informationen oder Betriebsmittel sollen bestimmten Parteien bei Bedarf zur Verfügung stehen. Um die unbefugte Vorenthaltung von Informationen oder Betriebsmitteln zu verhindern, können entsprechende Mechanismen eingesetzt werden, die jedoch nicht der Kryptographie zugeordnet werden. Hier können beispielsweise redundante Systeme verwendet werden [26].
- Authentizität (Authenticity): Sowohl Daten als auch Parteien, die miteinander kommunizieren, sollen auf ihre Echtheit hin geprüft werden können. Man unterscheidet dementsprechend zwischen der Authentizität des Datenursprungs (Data Origin Authenticity), die auch die Integrität der Daten beinhaltet, und der Authentizität der Parteien (Entity Authenticity). Um die Echtheit von Daten sicherzustellen, können Message Authentication Codes und digitale Signaturen eingesetzt werden. Mittels Authentifizierungsverfahren (auch: Authentifizierungsprotokollen) kann nachgewiesen werden, daß die miteinander kommunizierenden Parteien die sind, für die sie sich ausgeben (Nachweis der Identität).
- Nichtabstreitbarkeit (Non-Repudiation): Mit Nichtabstreitbarkeitsmechanismen soll gegenüber Beteiligten und Unbeteiligten bewiesen werden, ob ein bestimmtes Ereignis eingetreten ist bzw. eine bestimmte Aktion ausgeführt wurde oder nicht. Das

Ereignis oder die Aktion kann dabei das Erzeugen, das Übermitteln, die Entgegennahme oder das Vorlegen einer Nachricht sein. Non-Repudiation-Zertifikate, -Tokens und -Protokolle ermöglichen die Verbindlichkeit von Daten. Die verwendeten Mechanismen basieren auf Message Authentication Codes oder digitalen Signaturen in Verbindung mit den Diensten Notary Services, Timestamping Services und Evidence Recording.

Um die Vertrauenswürdigkeit von IT-Systemen meßbar und damit vergleichbar zu machen, wurden in vielen Ländern Kataloge für Sicherheitskriterien ausgearbeitet [2, 8, 18, 19, 21]. Beispielhaft sei der europaweit geltende ITSEC-Kriterienkatalog [8] erwähnt, der Kriterien zur Evaluierung der Sicherheit von IT-Systemen beinhaltet. Er definiert Sicherheitskriterien in unterschiedlichen Stufen zu den folgenden drei grundsätzlichen Bedrohungen, die sich aus dem Verlust der von einem IT-System geforderten Sicherheit ableiten lassen:

- Bedrohung der Vertraulichkeit (unbefugte Preisgabe von Informationen),
- Bedrohung der Integrität (unbefugte Veränderung von Daten),
- Bedrohung der Verfügbarkeit (unbefugte Vorenthaltung von Informationen oder Betriebsmitteln).

Gegenstand der nachstehenden Ausführungen sind die heute für IT-Systeme existierenden grundlegenden Sicherheitsmechanismen, mit denen die genannten Sicherheitsanforderungen erfüllt werden können. Bei den als Sicherheitsmechanismen beschriebenen Verfahren, wie Verschlüsselungsverfahren, Authentifizierungsverfahren und digitalen Signaturverfahren, werden kryptographische Mechanismen eingesetzt, von denen die wichtigsten im folgenden näher erläutert werden.

3 Kryptographische Mechanismen

Moderne kryptographische Mechanismen basieren hauptsächlich auf verschiedenen, nicht bewiesenen Annahmen der Komplexitätstheorie über die „leichte“ und „schwere“ Berechenbarkeit von Funktionen durch Algorithmen. Dabei steht der Begriff der „leichten“ Berechenbarkeit im Gegensatz zum Begriff der „schweren“ Berechenbarkeit. Intuitiv versteht man unter „leicht“ bzw. „schwer“ zu berechnenden Funktionen folgendes: Man sagt, eine Funktion $f: X \rightarrow Y$ ist leicht zu berechnen, wenn ein praktisch durchführbarer Algorithmus bekannt ist, der für beliebiges $x \in X$ den Wert $f(x)$ berechnet. Man sagt, eine Funktion $f: X \rightarrow Y$ ist schwer zu berechnen, wenn kein praktikabler Algorithmus „bekannt“ ist, der auch nur für einen nennenswerten Teil der Elemente von X den Funktionswert $f(x)$ berechnet, und wenn es Gründe gibt anzunehmen, daß ein solcher Algorithmus auch „nicht gefunden“ werden kann. Dies sind im mathematischen Sinne sicherlich keine exakten Definitionen, da man „bekannt“ und „nicht gefunden“ nicht näher definieren kann.

Es gibt Versuche, die intuitiven Vorstellungen über leicht und schwer zu berechnende Funktionen mathematisch zu präzisieren. Dies erweist sich aber als äußerst schwierig und führt auf ungelöste Probleme der Logik und Algorithmentheorie. Erste Ansätze, die Begriffe leicht und schwer zu berechnende Funktionen in Definitionen zu fassen, liefert die Komplexitätstheorie mit Untersuchungen des Rechenzeit- und Speicherplatzbedarfs eines Algorithmus in Abhängigkeit vom Umfang der Eingabedaten.

In der Kryptographie werden insbesondere die zwei folgenden Funktionenklassen verwendet: Die One-Way-Funktionen und die Trapdoor-One-Way-Funktionen.

- Eine One-Way-Funktion $f: X \rightarrow Y$ besitzt die Eigenschaft, daß es zu einem gegebenen x leicht ist, den Funktionswert $f(x)$ zu berechnen, es aber für nahezu alle Elemente $y \in Bild(f)$ schwer ist, ein $x \in X$ zu finden, so daß $f(x)=y$ gilt.
  Zu dieser Funktionenklasse zählt beispielsweise die folgende Funktion: Für $f: P \times P \rightarrow \mathbb{N}$ mit $(p,q) \rightarrow p \cdot q$ kann das Produkt n zweier großer Primzahlen, deren Anzahl an Dezimalstellen größer als 100 ist, leicht berechnet werden. Es ist jedoch schwer, eine in dieser Größenordnung gegebene Zahl n in ihre Primfaktoren zu zerlegen. Dieses Problem wird Faktorisierungsproblem genannt. Eine weitere One-Way-Funktion ist die in Abschnitt 5 beschriebene Hashfunktion.
- Eine Trapdoor-One-Way-Funktion ist eine One-Way-Funktion $f: X \rightarrow Y$ mit der zusätzlichen Eigenschaft, daß es mit Kenntnis einer bestimmten Zusatzinformation (Trapdoor-Information genannt) leicht ist, für jedes beliebige Element $y \in Bild(f)$ ein $x \in X$ zu finden, so daß $f(x)=y$ gilt.
  Im Beispiel des Faktorisierungsproblems repräsentieren die Primfaktoren von n die Trapdoor-Information.

Die wichtigsten kryptographischen Mechanismen werden mit Hilfe von Kryptosystemen realisiert. Kryptosysteme bestehen aus zwei Mengen an Funktionen, einer Menge an Schlüsseln, durch die diese Funktionen parametrisiert werden, und aus Mengen, auf denen diese Funktionen operieren. Man unterscheidet zwischen symmetrischen Kryptosystemen (oder Private-Key-Kryptosystemen) und asymmetrischen Kryptosystemen (oder Public-Key-Kryptosystemen).

Bei einem symmetrischen Kryptosystem besitzen die jeweils miteinander kommunizierenden Parteien den gleichen Schlüssel K. Dieser Schlüssel muß bei den rechtmäßigen Besitzern streng geheimgehalten werden. Er wird aufgrund dieser Anforderung geheimer Schlüssel genannt und stellt ein Geheimnis dar, das nur den miteinander kommunizierenden Parteien bekannt ist. Der Schlüsselraum, aus dem K gewählt wird, muß dabei so groß sein, daß es schwer ist, beispielsweise durch vollständige Suche auf den Schlüssel K schließen zu können.

Asymmetrische Kryptosysteme basieren auf Trapdoor-One-Way-Funktionen. Jede Partei besitzt ein Schlüsselpaar (PK,SK), das aus einem geheimzuhaltenden Schlüssel SK (geheimer Schlüssel) besteht und dem zu diesem gehörenden Schlüssel PK, der veröffentlicht werden kann (öffentlicher Schlüssel). Der öffentliche

Schlüssel PK jeder Partei kann beispielsweise in einem öffentlichen Schlüsselverzeichnis (Directory), ähnlich einem Telefonbuch, publiziert werden. Der geheime Schlüssel SK unterliegt strengster Geheimhaltung und repräsentiert ein Geheimnis, das nur sein rechtmäßiger Besitzer kennt. Das Schlüsselpaar (SK,PK) besitzt die Eigenschaft, daß es schwer ist, SK aus PK zu berechnen, mit Kenntnis der Trapdoor-Information jedoch leicht.

Die Kryptographie beruht neben der Annahme über die Berechenbarkeit von Funktionen auf der Voraussetzung, daß Daten, die für kryptographische Mechanismen eingesetzt werden, authentisch sind (beispielsweise durch ihre Veröffentlichung) oder ihre Authentizität geprüft werden kann. Ein Datum oder eine Partei gilt dann als echt, wenn ein Prüfer einen Beweis akzeptiert, in dem ein Geheimnis verwendet wird, bei dem der Prüfer davon ausgeht, daß dieses Geheimnis nur dem rechtmäßigen Besitzer oder den rechtmäßigen Besitzern bekannt ist. Jeder andere, der das Geheimnis kennt, also auch ein Betrüger, kann das Geheimnis ebenfalls für einen entsprechenden Beweis einsetzen. Es ist Aufgabe der Sicherheitsstrategie (Security Policy) der jeweiligen Anwendung festzulegen, ob ein Prüfer einen Beweis akzeptieren kann. Die Sicherheitsstrategie muß definieren, wie hoch das Sicherheitsniveau der Anwendung zu setzen ist, beispielsweise: wie streng die Auflagen an die Sicherheit von Geheimnissen (wie Größe, Anzahl aber auch Aufbewahrungsort) sind und mit welchem Verfahren der Sicherheitsmechanismus realisiert wird, damit er als sicher im Sinne der Sicherheitsstrategie gilt. Hier müssen Kosten, die durch den Einsatz von Sicherheitsmechanismen entstehen, und Kosten, die durch Schäden ohne oder von in nur geringem Maße eingesetzten Sicherheitsmechanismen aufkommen, gegeneinander abgewogen werden.

Bei den im folgenden beschriebenen Verfahren wird vorausgesetzt, daß alle Berechnungen in einer sicheren Umgebung durchgeführt und die Daten unverändert über den Kommunikationskanal gesendet werden.

4 Vertraulichkeit

Vertraulichkeit kann durch Verschlüsselungsverfahren erreicht werden. Diese Verfahren werden verwendet, um Informationen gegenüber unerwünschten Parteien geheimzuhalten (Wahrung der Vertraulichkeit).

Bei diesen Verfahren werden Funktionen eingesetzt, die durch einen Schlüssel K1 aus dem Schlüsselraum parametrisiert sind (Verschlüsselungsfunktionen), und Funktionen, die durch einen Schlüssel K2 aus dem Schlüsselraum parametrisiert sind (Entschlüsselungsfunktionen). Die Funktionen besitzen die Eigenschaft, daß zu jedem Schlüssel K1 aus dem Schlüsselraum ein eindeutiger Schlüssel K2 aus dem Schlüsselraum existiert, derart, daß Verschlüsselungsfunktion und Entschlüsselungsfunktion invers zueinander sind.

Die zu verschlüsselnden Daten werden mittels der durch K1 parametrisierten Verschlüsselungsfunktion transformiert. Aus dem so erzeugten Chiffrat können bei Kenntnis des Schlüssels K2 mittels der zur Verschlüsselungsfunktion inversen Entschlüsselungsfunktion die Daten zurückgewonnen werden.

Symmetrische und einige asymmetrische Kryptosysteme lassen sich als Verschlüsselungsverfahren verwenden. Darüber hinaus gibt es sogenannte Session-Key-Verfahren (oft auch hybride Kryptosysteme genannt), die beide Kryptosysteme einsetzen. Da das Session-Key-Verfahren zu einem der wichtigsten und verbreitetsten Verschlüsselungsverfahren gehört, wird es im folgenden vorgestellt.

Bei einem Session-Key-Verfahren wird sowohl ein symmetrisches als auch ein asymmetrisches Kryptosystem als Verschlüsselungsverfahren eingesetzt (siehe Abbildung 1. x||y bezeichnet dabei die Konkatenation von x und y, // symbolisiert den Kommunikationskanal).

Der Klartext m wird mit einem symmetrischen Kryptosystem verschlüsselt. Der geheime Schlüssel (Session Key K) für diese Verschlüsselung wird von Partei A vor Beginn einer jeden Kommunikation (Session) in Form einer Pseudozufallszahl erzeugt. Eine Pseudozufallszahl ist eine Zahl, die Element einer reproduzierbaren Zahlenfolge ist und mittels deterministischer Algorithmen ausgehend von einem echt zufällig gewählten Startwert berechnet wird. Diese Folge besitzt die Eigenschaft, daß es schwer ist, aus der Kenntnis einiger Zufallszahlen die als nächstes erzeugten Zufallszahlen vorherzusagen. Der so gebildete Session Key wird aus Sicherheitsgründen nur für eine einzige Sitzung verwendet.

Partei A (Sender) verschlüsselt nun mit der Verschlüsselungsfunktion, der durch den Schlüssel K parametrisierten Funktion E_1, den Klartext m. Für die sichere Übermittlung des geheimen Schlüssels K an Partei B (Empfänger) wird ein asymmetrisches Kryptosystem eingesetzt: Der Session Key K wird mit der Verschlüsselungsfunktion, der durch den öffentlichen Schlüssel PKB des Empfängers parametrisierten Funktion E_2, verschlüsselt (Schlüsselchiffrat c_K). Das Schlüsselchiffrat c_K wird dann zusammen mit dem verschlüsselten Klartext c_m an B übermittelt.

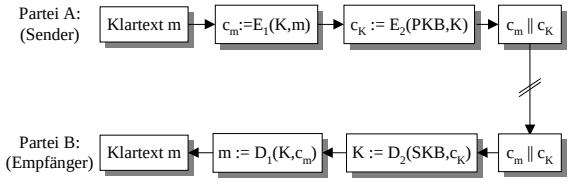

Abbildung 1: Session-Key-Verfahren

Partei B (Empfänger) gewinnt den Session Key K zurück, indem sie zunächst das Schlüsselchiffrat c_K entschlüsselt: basierend auf dem asymmetrischen Kryptosystem erhält sie mit Hilfe der Funktion D_2, die mit dem zum öffentlichen Schlüssel PKB gehörenden geheimen Schlüssel SKB parametrisiert ist, den Schlüssel K. Mit Hilfe von K berechnet B mit der Funktion D_1 des symmetrischen Kryptosystems aus den verschlüsselten Daten c_m den Klartext m.

Das Session-Key-Verfahren ist insbesondere für die Verschlüsselung von großen Datenmengen geeignet, da die Performanceeigenschaften der beiden Kryptosysteme gewinnbringend genutzt werden können. Beispiels-

weise besitzt das bekannteste symmetrische Kryptosystem, der DES (Data Encryption Standard [16]), eine wesentlich bessere Performance als das bekannteste asymmetrische Kryptosystem, das RSA-Verfahren (benannt nach seinen Erfindern Rivest, Shamir, Adleman [24]): DES ist in Hardware ungefähr 1000 mal und in Software ungefähr 100 mal schneller als RSA [25].
Durch die hier beschriebene Kombination von symmetrischem und asymmetrischem Kryptosystem wird auch das Schlüsselaustauschproblem für die in symmetrischen Kryptosystemen eingesetzten geheimen Schlüssel behoben. Es müssen lediglich die öffentlichen Schlüssel des asymmetrischen Kryptosystems authentisch ausgetauscht werden. Eine mögliche Lösung für die Gewährleistung der Authentizität von öffentlichen Schlüsseln wird in Abschnitt 9 vorgestellt. Durch die Verschlüsselung von K mittels PKB kann nur der Besitzer des geheimen Schlüssels SKB den zur Verschlüsselung eingesetzten geheimen Schlüssel K und damit den Klartext m wiedergewinnen.
Beispiele für symmetrische Kryptosysteme, die als Verschlüsselungsverfahren verwendet werden, sind: DES [16], Triple-DES [1] und IDEA [15]. Beispiele für asymmetrische Kryptosysteme, die als Verschlüsselungsverfahren eingesetzt werden, sind: RSA [24] und ElGamal [5]. Als Session-Key-Verfahren sind die Kombinationen DES mit RSA oder IDEA mit RSA üblich.

5 Datenintegrität

Die Integrität (Unversehrtheit) von Daten kann mit sogenannten One-Way-Hashfunktionen (kurz: Hashfunktionen) überprüft werden. Diese Funktionen werden oftmals auch als Manipulation Detection Code (MDC), Message Digest, Digital Fingerprint, kryptographische Prüfsumme oder Message Integrity Check (MIC) bezeichnet. Es handelt sich hierbei um einen Mechanismus, der Manipulationen an Daten nicht verhindern, sie aber im nachhinein erkennbar machen kann (Detektionsmechanismus). Die gesicherten Daten liegen unverändert im Klartext vor.
Eine Hashfunktion H ist eine One-Way-Funktion, die die in Abschnitt 3 beschriebene Eigenschaft besitzt, daß ihr Funktionswert H(m) „leicht“ zu berechnen ist, es aber „schwer“ ist, von H(m) auf das Argument m zu schließen. Die Hashfunktion H bildet eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit einer maximalen oder einer festen Länge ab. Betrachtet man Binärwerte als Eingabewerte, so kann H wie folgt definiert werden: $H: \{0,1\}^* \rightarrow \{0,1\}^n$, wobei n typischerweise die Werte 64, 128 oder 160 annimmt. Eine Hashfunktion reduziert die Daten m auf ihren sogenannten Hashwert h := H(m).
Da es unendlich viele Zeichenfolgen mit beliebiger Länge gibt, aber nur endlich viele mit einer Länge $\leq n$, werden verschiedene Eingabewerte auf den gleichen Hashwert abgebildet. Dies bezeichnet man als Kollision. Hashfunktionen müssen kollisionsresistent sein: Es darf praktisch nicht möglich sein, zwei verschiedene Zeichenfolgen m_1 und m_2 zu finden, die den gleichen Hashwert $H(m_1) = H(m_2)$ besitzen.

Die Verwendung einer Hashfunktion zeigt Abbildung 2.

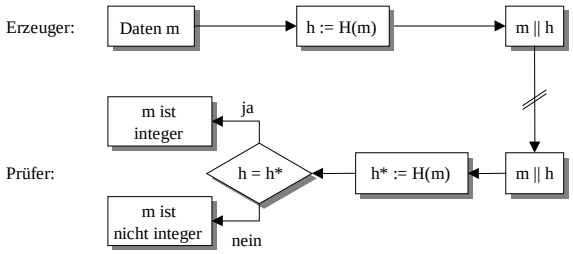

Abbildung 2: Hashfunktion H

Die Integrität der Daten m wird geprüft, indem man den zuerst erzeugten Hashwert h mit einem neu erzeugten Hashwert h* vergleicht. Stimmt h mit h* überein, gelten die Daten (und auch der Hashwert) als integer. Die Veränderung eines Bits in der Eingabemenge m oder dem Hashwert h führt dazu, daß die Daten als nicht integer gelten. Hashfunktionen sollten darüber hinaus die Eigenschaft besitzen, daß sich die Veränderung eines einzelnen Bits in der eingegebenen Zeichenfolge m auf 50% der Bits ihres Hashwertes h auswirkt (Avalanche-Effekt).
Hashfunktionen sind öffentlich, das bedeutet, man muß zur Berechnung ihres Funktionswertes keine Geheimnisse besitzen. Somit kann jeder, der die Funktion kennt, den Hashwert berechnen und damit die Integrität der Daten überprüfen.
Beispiele für Hashfunktionen sind: MD5 [23], RIPEMD-128 [4], RIPEMD-160 [4] und SHA-1 [17].

6 Authentizität des Datenursprungs

Mit den beiden folgenden Verfahren kann neben der Integrität der Daten auch die Authentizität (Echtheit) des Datenursprungs geprüft werden:

- Message Authentication Code (MAC) und
- digitale Signatur.

Wie bei den Verfahren zur Datenintegrität handelt es sich um Detektionsmechanismen. Die gesicherten Daten liegen ebenfalls unverändert im Klartext vor.

6.1 Message Authentication Code

Der Message Authentication Code (MAC) ist eine Hashfunktion MAC := h = H(k,m), die zusätzlich von einem geheimen Schlüssel k abhängig ist. Die Sicherheit des MACs hängt außer von der Länge des erzeugten Hashwertes auch von der Sicherheit des verwendeten Schlüssels k ab. Nur derjenige, der den geheimen Schlüssel k (und die Funktion H) kennt, kann den MAC berechnen.
Ein MAC wird wie folgt verwendet (siehe Abbildung 3):
Der Erzeuger will seine Daten m durch einen MAC sichern. Hierzu berechnet er mit einer Hashfunktion, die durch einen geheimen Schlüssel k parametrisiert ist, zu seinen Daten die Prüfsumme: MAC := H(k,m). Jeder, der über den gleichen geheimen Schlüssel k (und Hashfunktion H) verfügt wie der Erzeuger des MACs, kann nun die Daten m (und auch den MAC) auf Au-

thentizität prüfen. Hierzu berechnet der Prüfer zunächst zu den Daten m eine Prüfsumme MAC* := H(k,m). Anschließend führt er eine Ja/Nein-Entscheidung aus: Stimmt der ursprüngliche MAC mit dem neu berechneten MAC* überein, dann gelten die Daten (und auch der MAC) als authentisch. Stimmen die beiden Werte nicht überein, dann können entweder die Daten m oder der MAC zwischen dem Zeitpunkt der MAC-Bildung und seiner Prüfung verändert worden sein oder es wurden unterschiedliche Schlüssel verwendet.

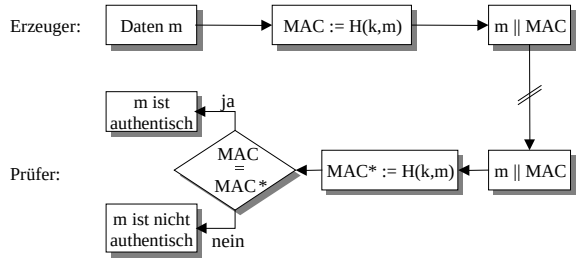

Abbildung 3: Message Authentication Code (MAC)

Bei diesem Verfahren ist zu beachten, daß der Prüfer den gleichen MAC erzeugen kann wie der Erzeuger, da er über den gleichen geheimen Schlüssel k verfügt – ja sogar verfügen muß. Demnach besitzt ein MAC keine Beweiskraft gegenüber Dritten.
Ein einfacher MAC ergibt sich beispielsweise durch die Verwendung einer Blockchiffre, die im Betriebsmodus CBC (Cipher Block Chaining) arbeitet (siehe Abbildung 4).

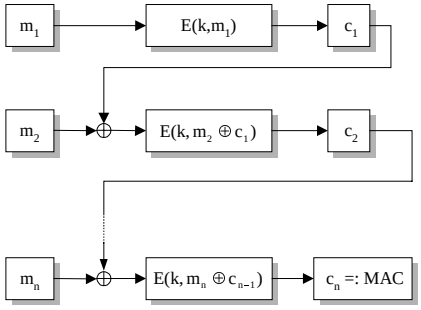

Abbildung 4: CBC-basierter MAC

Die Daten m werden in n Blöcke gleicher Länge aufgeteilt: $m = m_1||m_2||...||m_n$, wobei die Länge durch die Blockchiffre vorgeschrieben ist (beispielsweise 64-Bit-Blöcke). Gegebenenfalls muß der letzte Block m_n durch geeignete Bits auf Blocklänge aufgefüllt werden (Padding). Jeder Block m_i wird mit dem vorhergehend erzeugten Chiffretextblock c_{i-1} (i>1) verknüpft und mit der Verschlüsselungsfunktion E, die durch den geheimen Schlüssel k parametrisiert ist, verschlüsselt. Der letzte Chiffretextblock c_n (oftmals auch nur ein Teil des Chiffretextblockes) bildet dann den MAC zu den Daten m.
Ist der Schlüssel k öffentlich bekannt, so kann die Hashfunktion als Manipulation Detection Code aufgefaßt werden.

6.2 Digitale Signatur

Die Idee und der Begriff der digitalen Signatur erscheinen zum ersten Mal als „digital signature“ bei Diffie und Hellman [3]. Sie schlagen vor: Die digitale Signatur einer Partei A zu ihren Daten m soll ein Wert sein, der von m und von einer zusätzlichen, geheimen Information abhängt, die nur A bekannt ist. Jeder Benutzer des digitalen Signaturverfahrens kann die Echtheit der von A erstellten Signatur überprüfen (Verifikation), indem er eine weitere, bekanntgegebene, also öffentliche Information von A benutzt. Wie gesichert werden kann, daß diese Information tatsächlich zu A gehört, wird in Abschnitt 9 vorgestellt. Da nur A im Besitz der geheimen Information ist, kann nur sie mit der Signierfunktion S diese digitale Signatur zu den Daten m erstellen. Somit besitzt die digitale Signatur im Gegensatz zu einem MAC auch Beweiskraft gegenüber Dritten.
Als digitale Signaturverfahren eignen sich asymmetrische Kryptosysteme, da sie Trapdoor-One-Way-Funktionen verwenden. Es gibt auch auf symmetrischen Kryptosystemen basierende Signaturverfahren. Da diese jedoch nicht sehr anwendungsfreundlich sind, werden sie hier nicht betrachtet.
Beispiele für asymmetrische Kryptosysteme, die als digitale Signaturverfahren eingesetzt werden können, sind: RSA [24], DSS [20], ElGamal [5], GMR [7] und Fiat-Shamir [6].
Bei digitalen Signaturverfahren darf das zu signierende Dokument m eine bestimmte Größe, die durch den Zahlenraum des jeweiligen digitalen Signaturverfahrens bestimmt wird, nicht überschreiten. Die durch Schlüssel parametrisierten Funktionen der digitalen Signaturverfahren operieren beispielsweise auf den endlichen Zahlenräumen $\mathbb{Z}_n \rightarrow \mathbb{Z}_n$ mit $n = p \cdot q$ oder $GF(p) \rightarrow GF(p)$ mit p und q prim. Um trotzdem zu Daten, die außerhalb der Zahlenräume liegen (im Beispiel $m \geq n$ oder $m \geq p$) digitale Signaturen erstellen und verifizieren zu können, sind zwei Möglichkeiten denkbar: Bei der ersten Möglichkeit werden die Daten m in Blöcke $m_1,...,m_k$ mit beispielsweise $m_i < n$ aufgeteilt und jeder Block einzeln signiert. Bei der zweiten, in der Praxis üblichen Methode wird eine Kopie von m mittels einer Hashfunktion auf einen Wert H(m)<n reduziert, der dann signiert wird. Die Signatur wird damit nicht aus den Daten selbst, sondern aus dem Hashwert der Daten berechnet. Diese Methode erhöht sowohl die Performance als auch die Sicherheit, beispielsweise kann die Reihenfolge der signierten Daten nicht mehr unbemerkt verändert werden.
Hashfunktionen, die in Zusammenhang mit Signaturverfahren verwendet werden, sind beispielsweise MD5 [23], RIPE-MD 128 [4], RIPE-MD 160 [4] und SHA-1 [17].
Um die Authentizität von Daten mit Hilfe von digitalen Signaturverfahren zu sichern, geht man in der Praxis wie folgt vor (siehe Abbildung 5). Die Beschreibung beschränkt sich hierbei auf die einfachste Ausführung eines digitalen Signaturverfahrens (beispielsweise RSA [24]).
Signierer A will seine Daten m unter Verwendung einer Hashfunktion H digital signiert an einen Verifizierer

übermitteln. A bildet zu m mit Hilfe der Hashfunktion H den Hashwert h := H(m). Anschließend berechnet A mit der Signierfunktion S, die durch einen nur ihm bekannten Wert, seinem geheimen Schlüssel SKA, parametrisiert ist, den Wert s := S(SKA,h). A sendet seine Daten m mit der dazugehörigen digitalen Signatur s an den Verifizierer. Das Konkatenat m||s wird dabei als signierte Nachricht bezeichnet.

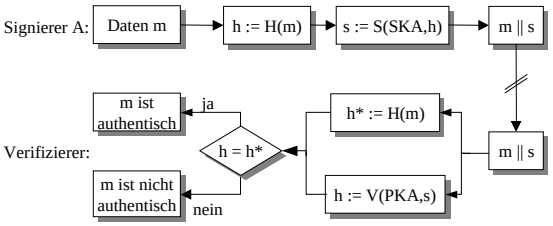

Abbildung 5: Das Prinzip einer digitalen Signatur

Der Verifizierer benötigt zur Prüfung der digitalen Signatur s den öffentlichen Schlüssel PKA von A, die von A benutzte Hashfunktion H und die Verifikationsfunktion V. Er bildet zunächst einen Hashwert h* := H(m). Mit Hilfe von V, parametrisiert durch den öffentlichen Schlüssel PKA, entscheidet er nun, ob die Signatur zu m korrekt und m damit unverändert ist oder nicht: Er berechnet aus der Signatur s den Hashwert h := V(PKA,s) und prüft, ob h = h* gilt. Bei Gleichheit gilt die Signatur von A als korrekt und die Daten m als unverändert. Da nur A im Besitz des geheimen Schlüssels SKA ist, kann nur er eine korrekte Signatur s zu m bilden. Ist $h \neq h^*$, so gilt die Signatur als falsch und die Daten als nicht authentisch. Dies kann beispielsweise durch die Veränderung der Daten m oder der Signatur s zwischen dem Zeitpunkt des Signierens und Verifizierens verursacht worden sein. Es kann aber auch der öffentliche Schlüssel falsch sein, beispielsweise dadurch, daß er nicht dem Signierer gehört.

Neben diesen relativ einfachen Möglichkeiten zur Berechnung und Prüfung der Signatur (Signature With Appendix [11]) gibt es weitere, komplexere Methoden, die die Signaturbildung selbst betreffen (wie Signature Given Message Recovery oder Signature Given Limited Message Recovery [14]).

Sollen die Daten sowohl vertraulich als auch authentisch übermittelt werden, dann signiert der Sender zunächst die Daten mittels seines geheimen Schlüssels und verschlüsselt die signierte Nachricht, indem er den öffentlichen Schlüssel des Empfängers verwendet.

7 Authentizität der Parteien

Wie im vorigen Abschnitt erläutert wurde, kann mit Hilfe von digitalen Signaturverfahren die Authentizität des Datenursprungs nachgewiesen werden. Darüber hinaus erweist es sich häufig als notwendig, auch die Authentizität der Parteien zu gewährleisten, beispielsweise wenn sichergestellt werden muß, daß die miteinander kommunizierenden Parteien auch die sind, für die sie sich ausgeben. Verfahren, die für das Erbringen eines solchen Nachweises eingesetzt werden, bezeichnet man als Authentifizierungsverfahren (auch: Authentifizierungsprotokolle). In den Daten, die während des Protokollablaufs zwischen den Parteien übermittelt werden, können auch Textfelder enthalten sein, über die beispielsweise geheime Schlüssel für die weitere vertrauliche Kommunikation ausgetauscht werden können.

Im folgenden wird die einfachste Variante eines Authentifizierungsverfahrens vorgestellt: das Challenge& Response-Verfahren. Dieses Verfahren kann auf der Basis eines symmetrischen oder eines asymmetrischen Kryptosystems (siehe Abbildungen 6, 7 und 8) realisiert werden [12].

Das Prinzip eines Challenge&Response-Verfahrens läßt sich wie folgt beschreiben: Der Prüfer übermittelt an den Beweiser eine zufällig erzeugte Zahl R, die sogenannte Challenge. Der Beweiser sendet eine Antwort (Response) an den Prüfer, die eine Zahl enthält, die unter Verwendung von R und einem Geheimnis erzeugt wurde. Der Prüfer prüft nun, ob die empfangene Zahl aus dem von ihm erzeugten R und dem Geheimnis, von dem er annimmt, das es der zu Prüfende besitzt, berechnet wurde. Wenn der Prüfer dies verifizieren konnte, dann gilt der Beweiser als authentisch. Für jeden Authentifizierungsvorgang wird eine neue Challenge erzeugt, weshalb diese Art der Authentifizierung auch dynamische Authentifizierung genannt wird.

Bei der Authentifizierung wird zwischen einseitiger und gegenseitiger Authentifizierung unterschieden. Bei der einseitigen Authentifizierung beweist eine Partei der anderen Partei ihre Authentizität, bei der gegenseitigen Authentifizierung beweisen sich beide Parteien ihre Authentizität gegenseitig. Beide Verfahren werden im folgenden erläutert.

Bei einem **Challenge&Response-Verfahren auf der Basis eines symmetrischen Kryptosystems** müssen die beiden Parteien über den gleichen geheimen Schlüssel K verfügen. Nachstehend wird die einseitige Authentifizierung nach ISO 9798-2 beschrieben (siehe Abbildung 6).

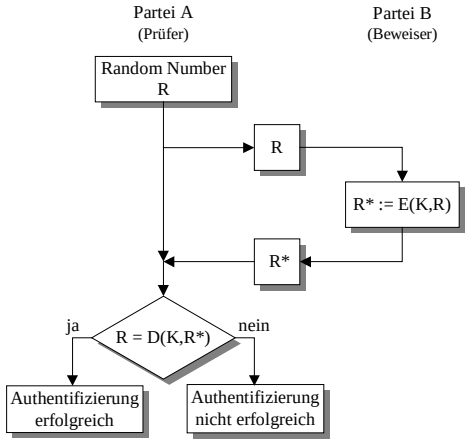

Abbildung 6: Einseitige Authentifizierung auf der Basis eines symmetrischen Kryptosystems

Partei A (Prüfer) möchte sich von der Identität der Partei B (Beweiser) überzeugen. Hierzu erzeugt A eine Zufallszahl R (Challenge) und sendet R an B. Der zu Prüfende, Partei B, verschlüsselt diese Zufallszahl mit seiner Verschlüsselungsfunktion E, parametrisiert durch

den geheimen Schlüssel K. Er sendet das so erzeugte Chiffrat R* (Response) an A. Partei A entschlüsselt mit der ebenfalls durch den Schlüssel K parametrisierten Funktion D das erhaltene Chiffrat und prüft, ob der berechnete Wert mit der von ihr gesendeten Zufallszahl R übereinstimmt. Bei Übereinstimmung geht A davon aus, daß B zur Berechnung von R* das Geheimnis K, das A mit B teilt, verwendet hat, und akzeptiert B als authentisch.

Da bei diesem Verfahren jeder Kommunikationspartner den gleichen Schlüssel besitzt, ergeben sich hieraus hohe Sicherheitsauflagen an die Schlüsselhaltung. Das Speichern von gleichen Schlüsseln kann beispielsweise durch sogenannte abgeleitete Schlüsselkonzepte verhindert werden: Aus einem Master Key und zusätzlichen Daten werden sogenannte individuelle Schlüssel abgeleitet, die dann im Challenge&Response-Verfahren eingesetzt werden. Angenommen, der Master Key MK ist bei Partei B gespeichert und Partei A verfügt über einen individuellen Schlüssel IK, den B aus MK berechnen kann. Hierzu übermittelt A Daten über seine Identität (IDA) an B, die dann als Funktionswert zur Berechnung des abgeleiteten Schlüssels IK dient: IK = f(MK,IDA). Damit verfügen sowohl A als auch B über einen gemeinsamen geheimen Schlüssel, der im Challenge&Response-Verfahren verwendet werden kann.

Sollen sich zwei Parteien gegenseitig authentifizieren, so gibt es hierzu zwei Möglichkeiten. Zum einen kann das vorgestellte einseitige Authentifizierungsverfahren zweimal hintereinander durchgeführt werden, wobei jeweils die Rolle von Beweiser und Prüfer getauscht wird. Um den Ablauf jedoch zu vereinfachen und um auch die Dauer der Authentifizierung zu reduzieren, wird zur gegenseitigen Authentifizierung das folgende Authentifizierungsverfahren verwendet (siehe Abbildung 7):

Partei A erzeugt eine Zufallszahl RA und fordert Partei B auf, ebenfalls eine Zufallszahl, RB, zu generieren und diese an sie zu übermitteln. A verschlüsselt das Konkatenat RA||RB und sendet das Chiffrat E(K,RA||RB) an B. Partei B entschlüsselt das Chiffrat und prüft, ob RB der von ihr erzeugten Zufallszahl entspricht. Ist dies der Fall, dann verschlüsselt B das Konkatenat RB||RA und sendet das Chiffrat E(K,RB||RA) an A. Partei A entschlüsselt das Chiffrat und testet, ob RA die von ihr erzeugte Zufallszahl ist. Verlaufen beide Prüfungen erfolgreich, dann hat sich sowohl A gegenüber B als auch B gegenüber A authentifiziert.

Durch diesen Ablauf wird die Sicherheit beträchtlich erhöht, da es zwischen den einzelnen Daten, die übermittelt werden, eine überprüfbare Abhängigkeit gibt und somit keine Daten unbemerkt in das Protokoll eingefügt werden können.

Challenge&Response-Verfahren auf der Basis eines asymmetrischen Kryptosystem verwenden die Tatsache, daß zur Authentifizierung digitale Signaturverfahren eingesetzt werden können. Hier werden zwei unterschiedliche Schlüssel verwendet: der öffentliche und der geheime Schlüssel des Beweisers. Die auf einem asymmetrischen Kryptosystem basierende einseitige

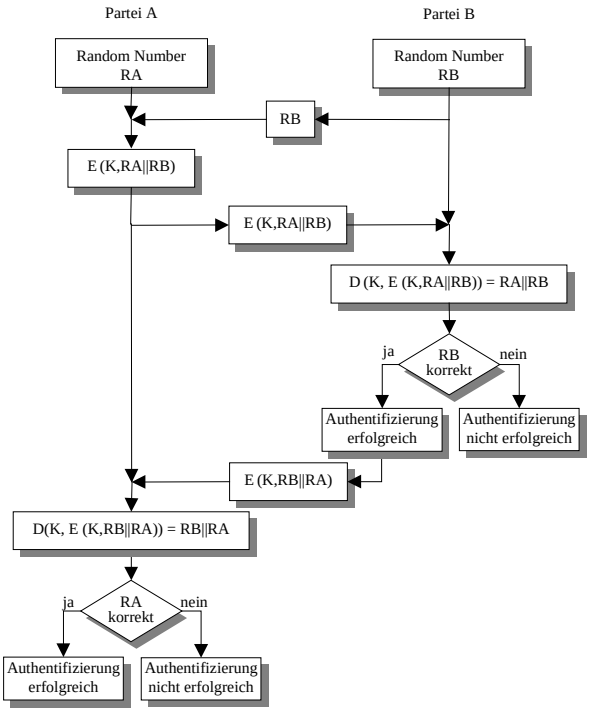

Abbildung 7: Gegenseitige Authentifizierung auf der Basis eines symmetrischen Kryptosystems

Authentifizierung läuft dann wie folgt ab (siehe Abbildung 8):

Partei A möchte sich von der Identität der Partei B überzeugen. Hierzu beschafft sich A den öffentlichen Schlüssel PKB der Partei B, beispielsweise aus einem öffentlichen Schlüsselverzeichnis. A erzeugt eine Zufallszahl R und sendet R über den Kommunikationskanal an B. Partei B signiert R mit ihrer Signierfunktion S, parametrisiert durch ihren geheimen Schlüssel SKB, und sendet das Resultat R* an A. A verifiziert mit V, parametrisiert durch den öffentlichen Schlüssel PKB, die erhaltene Signatur, indem sie prüft, ob der von ihr berechnete Wert mit R übereinstimmt. Bei Übereinstimmung geht A davon aus, daß B in Besitz des geheimen Schlüssels SKB und somit authentisch ist.

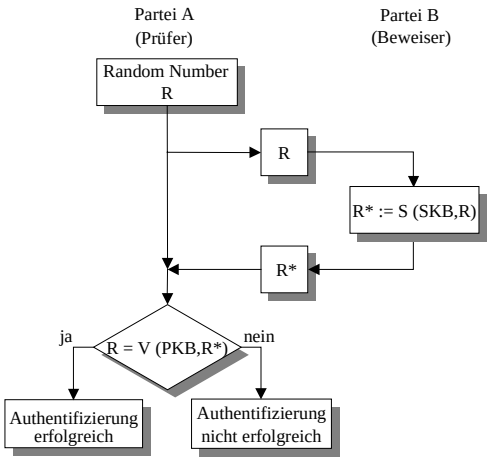

Abbildung 8: Einseitige Authentifizierung auf der Basis eines asymmetrischen Kryptosystems

Neben den beschriebenen, einfachen Protokollen gibt es weitere, komplexere Authentifizierungsprotokolle mittels Public-Key-Kryptosystemen, kryptographischer Prüfsummen und Zero-Knowledge-Techniken, die im Standard ISO/IEC 9798 [12] zu finden sind. Darin werden fünf Methoden beschrieben: die Authentifizierung in einem Schritt (Unilateral One Pass Authentication), die einseitige Authentifizierung in zwei Schritten (Unilateral Two Pass Authentication), die gegenseitige Authentifizierung in zwei Schritten (Mutual Two Pass Authentication), die gegenseitige Authentifizierung in drei Schritten (Mutual Three Pass Authentication) und die gegenseitige parallele Authentifizierung in zwei Schritten (Mutual Two Pass Parallel Authentication).
Im folgenden werden einige Anmerkungen zur Sicherheit von Challenge&Response-Verfahren gegeben. Wenn sowohl A als auch B die Kommunikation eröffnen können und außerdem die empfangene Zufallszahl ohne jegliche Prüfung als Challenge akzeptiert wird, dann ist der folgende Angriff, die sogenannte Replay-Attacke, denkbar: Prüfer A sendet an den Beweiser, der bei dieser Attacke von Angreifer X repräsentiert wird, eine Zufallszahl R1. Angreifer X eröffnet eine zweite Kommunikation und sendet, in der Rolle eines Prüfers, R1 an A zurück. A verschlüsselt nun als Beweiser die Zufallszahl R1 und sendet das Chiffrat R1* an X zurück. Damit ist die zweite Kommunikation beendet, und Angreifer X sendet, in der Rolle des Beweisers in der ersten Kommunikation, R1* an Prüfer A zurück, der die Kommunikation als authentisch anerkennen wird.
Um diesen (und andere mögliche) Angriffe zu verhindern, fügt man üblicherweise den übertragenen Daten die eindeutige Identifikationsnummer des Prüfers und/oder Beweisers hinzu [12]. Denkbar sind auch Zeitstempelverfahren, die jedoch das Problem aufweisen, daß A und B über synchronisierte Uhren verfügen müssen.

8 Nichtabstreitbarkeit

Digitale Signaturen alleine reichen nicht aus, um als Beweismittel für die Zurechenbarkeit von Daten und Handlungen zu ihrem Urheber anerkannt zu werden. Die beiden folgenden Beispiele können dies verdeutlichen:

- Eine Partei kann abstreiten, daß sie eine bestimmte Nachricht signiert hat, indem sie beispielsweise ihren geheimen Schlüssel anonym veröffentlicht und anschließend behauptet, der Schlüssel sei verloren gegangen oder wurde gestohlen. Damit kann sie auch behaupten, daß die Signatur zur Nachricht gefälscht ist.
- Eine Partei kann behaupten, daß Nachrichten, die von ihr bereits vor Bekanntgabe der Kompromittierung ihres geheimen Schlüssels signiert wurden, gefälscht sind. Damit kann sie auch bereits signierte Nachrichten mit einem früheren Zeitstempel versehen, sie erneut signieren und behaupten, die Signatur sei gefälscht.

Mit Hilfe von Sicherheitsinfrastrukturen und Sicherheitstechniken können hier Beweismittel bereitgestellt werden. Sogenannte Nichtabstreitbarkeit-Mechanismen [13], die auf symmetrischen Kryptosystemen (Message Authentication Codes) oder asymmetrischen Kryptosystemen (digitalen Signaturen) basieren, stellen solche Sicherheitstechniken zur Verfügung. Sie beinhalten Non-Repudiation-Zertifikate, -Tokens und -Protokolle. Vertrauenswürdige dritte Instanzen (Trusted Third Parties) werden zur Bereitstellung der Dienste Notary Services, Timestamping Services und Evidence Recording eingesetzt. Mit diesen Mechanismen kann gegenüber Beteiligten und Unbeteiligten bewiesen werden, ob ein bestimmtes Ereignis eingetreten ist bzw. eine bestimmte Aktion ausgeführt wurde oder nicht. Das Ereignis oder die Aktion kann das Erzeugen, das Senden, die Entgegennahme oder die Zustellung einer Nachricht sein.

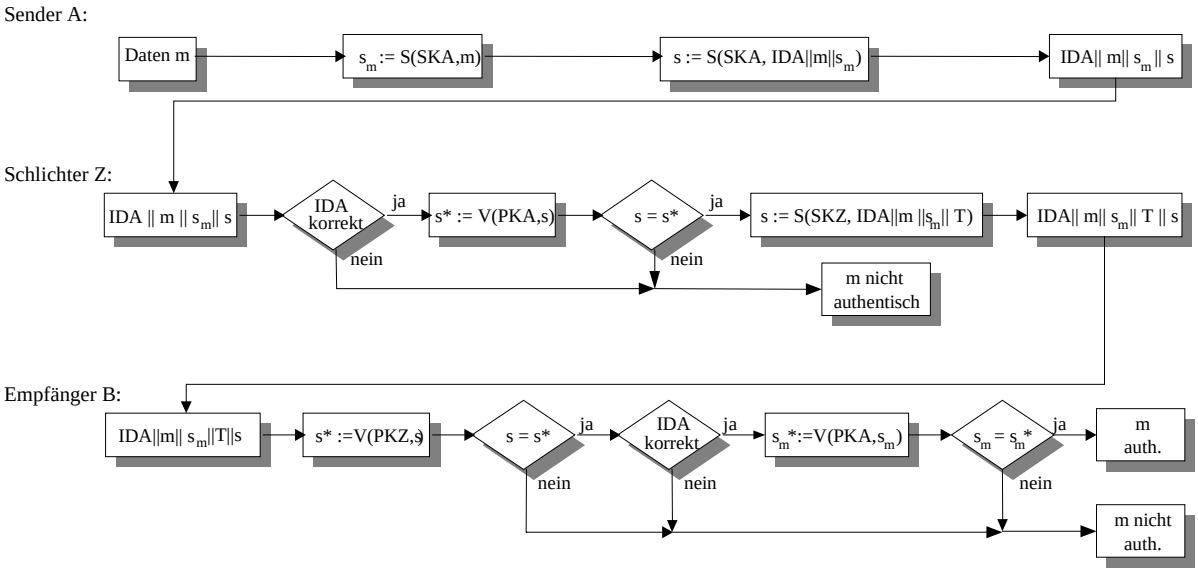

Abbildung 9: Geschlichtete digitale Signatur

Demnach werden die Mechanismen unterschieden in:

- Nichtabstreitbarkeit des Ursprungs (origin),
- Nichtabstreitbarkeit der Beförderung (transport).
- Nichtabstreitbarkeit der Lieferung (delivery),
- Nichtabstreitbarkeit der Vorlage (submission).

Im folgenden wird anhand von geschlichteten (arbitrated) digitalen Signaturen ein Beispiel für die Nichtabstreitbarkeit des Ursprungs gegeben (siehe Abbildung 9).

Partei A will an Partei B Daten übermitteln, wobei A im nachhinein nicht abstreiten können soll, daß sie die Daten erzeugt hat. A verfügt über einen Identitätsstring IDA, der eindeutig ihre Identität kennzeichnet. Sie signiert zunächst die Daten m mittels ihres geheimen Schlüssels SKA und fügt zu diesen Daten den String IDA hinzu. Anschließend signiert A das Konkatenat $IDA||m||s_m$ und übermittelt es zusammen mit der Signatur an eine vertrauenswürdige dritte Partei, den Schlichter (Arbiter) Z. Der Schlichter prüft zunächst IDA auf Korrektheit und verifiziert anschließend die Signatur s von A zu den Daten $IDA||m||s_m$. Verlaufen alle Prüfungen erfolgreich, dann versieht Z die Daten $IDA||m||s_m$ mit einem Zeitstempel T und signiert diese Sequenz. Die signierten Daten übermittelt er nun an B. Partei B verifiziert die Signatur von Z, prüft IDA auf Korrektheit und abschließend die Signatur s_m von A zu den Daten m. Verlaufen alle Prüfungen korrekt, dann kann A nicht abstreiten, die Daten erzeugt zu haben.

9 Public-Key-Infrastrukturen

Beim Einsatz von asymmetrischen Kryptosystemen ergeben sich folgende Probleme:

- Durch Session-Key-Verfahren kann erreicht werden, daß der verschlüsselte Session Key (und damit der Klartext) nur mit Kenntnis des geheimen Schlüssels des Empfängers wiedergewonnen werden kann (sogenannte adressierte Vertraulichkeit). Man kann jedoch nicht feststellen, ob der zum Verschlüsseln des Session Keys verwendete öffentliche Schlüssel tatsächlich zu einer bestimmten Partei gehört.
- Bei Signaturverfahren und signaturbasierten Authentifizierungsverfahren kann durch das Verifizieren der digitalen Signatur festgestellt werden, ob die Signatur zu bestimmten Daten mittels eines bestimmten Schlüssels erzeugt wurde. Nicht nachweisbar ist jedoch, ob der verwendete Schlüssel tatsächlich zu einer bestimmten Partei gehört.

Offensichtlich fehlt hier der authentische Zusammenhang zwischen dem öffentlichen Schlüssel und seinem Besitzer. Diese Verbindung kann beispielsweise durch sogenannte Public-Key-Zertifikate hergestellt werden [9, 10]. Für die Erzeugung von Zertifikaten ist eine vertrauenswürdige Instanz, ein sogenanntes Trust Center (TC), notwendig. Das Trust Center beglaubigt die Zusammengehörigkeit von Benutzern zu ihren öffentlichen Schlüsseln, darüber hinaus kann sie weitere Dienste erbringen wie Non-Repudiation-Service, Revocation Handling, Timestamping, Auditing und Directory Service. Innerhalb eines Trust Centers werden diese Leistungen von speziellen Komponenten erbracht. Jedes Trust Center, und damit auch seine Komponenten, arbeiten unter einer bestimmten Sicherheitsstrategie (Security Policy). Diese Strategie regelt beispielsweise, wie Zertifikate erstellt und herausgegeben werden, auch wie die Verfügbarkeit der Dienste gewährleistet wird.

10 Sicherheit für IT-Anwendungen

Ob die vorgestellten Sicherheitsmechanismen ohne weiteres für IT-Anwendungen verwendet werden können, muß speziell für jeden Einsatzfall geprüft werden. Aufgrund der Datenformate und der Datenmengen ergeben sich beispielsweise die folgenden Probleme:

- Aus Performancegründen kann es sinnvoll sein, die Daten nicht im Ganzen zu verschlüsseln, sondern vielmehr bestimmte Teile der Daten auszuwählen und nur diese zu verschlüsseln (partielle Verschlüsselung). Bei geeigneter Wahl kann damit eine ausreichende Vertraulichkeit der gesamten Datenmenge erreicht werden.
- Bei allen vorgestellten Sicherheitsmechanismen für Integrität und Authentizität darf kein einziges Bit der Eingabemenge verändert werden, sonst schlagen die Verifizierungen fehl. Will man beispielsweise die Authentizität von digitalen Bilddaten mittels digitaler Signatur sichern, so erweist es sich als schwierig, eine geeignete Eingabemenge für das Signaturverfahren zu definieren. Es müssen solche Daten sein, die erlaubte Operationen auf den Bildern wie Skalierung und Formatkonvertierung weiterhin zulassen, ohne daß die Verifizierung der Authentizität und Integrität der Daten fehlschlägt. Als Daten sind hier Merkmalsvektoren denkbar, die Bilddaten eindeutig kennzeichnen und die nicht durch zulässige Bildoperationen beeinflußt werden.
- Für Non-Repudiation-Dienste muß eine geeignete Sicherheitsinfrastruktur aufgebaut und eine zugehörige Sicherheitsstrategie definiert werden.

Auch die Sicherheitsanforderungen, die in Abschnitt 2 vorgestellt wurden, können das Gesamtspektrum an Sicherheitsanforderungen nicht abdecken. Eine weitere, essentielle Sicherheitsanforderung ist:

- Originalität von Daten: Es soll sichergestellt werden, daß die Daten unverändert und nicht in einer Kopie vorliegen.

Um unbefugtes Vervielfältigen von Daten erkennen zu können, werden ebenfalls Detektionsmechanismen eingesetzt, beispielsweise Copyright-Verfahren, digitale Wasserzeichen und Steganographie. Auch durch rechtliche Regelungen wie Urheberschutz, Patentschutz und Computerstrafrecht versucht man, dieser Bedrohung entgegenzuwirken.

11 Literatur

[1] ANSI X9.17 (Revised): American National Standard for Financial Institution Key Management (Wholesale). American Bankers Association, 1985.

[2] Department of Defense: Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). DOD 5200.28-STD, Dec 1985.

[3] Diffie, Whitfield; Hellman, Martin E.: New Directions in Cryptography. IEEE Transactions on Information Theory, Vol.22, Nr.6, 11/1976, pp.644-654.

[4] Dobbertin, Hans; Bosselaers, Antoon; Preneel, Bart: RIPEMD-160: A strengthened version of RIPEMD. Fast Software Encryption – Cambridge Workshop 1996, Md. 1039, Berlin: Springer 1996, pp.71-82.

[5] ElGamal, Taher: A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. IEEE Transactions on Information Theory, Vol.31, Nr.4, Jul 1985, pp.469-472.

[6] Fiat, Amos; Shamir, Adi: How to prove yourself: Practical solutions to identification and signature problems. Advances in Cryptology – Crypto'86 Proceedings, LNCS 263, Springer 1997, pp.186-194.

[7] Goldwasser, Shafi; Micali, Silvio; Rivest, Ronald L.: A 'Paradoxical' Solution to the Signature Problem. 25th Symposium on Foundations of Computer Science (FOCS), 1984, pp.441-448.

[8] Information Technology Security Evaluation Criteria (ITSEC): Provisional Harmonised Criteria. Version 1.2, Jun 1991.

[9] ISO/IEC 9594-8 | ITU-T Recommendation X.509: Information technology – Open Systems Interconnection – The Directory. Part 8: Authentication Framework, 1993.

[10] ISO/IEC 9594-8 | ITU-T Recommendation X.509: Final Text of Draft Amendments DAM 1 to ITU-T Recommendation X.509 (1993) | ISO/IEC 9594-8 on Certificate Extensions: ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7. Dec 1996.

[11] ISO/IEC 9796:1991 Information technology – Security techniques – Digital signature scheme giving message recovery.

[12] ISO/IEC 9798: Information technology – Security techniques – Entity authentication. Part 1 - Part 5.

[13] ISO/IEC 13888: Information technology – Security techniques – Non-Repudiation.

[14] Part 1: General (IS 1997). Part 2: Using symmetric techniques (DIS 1997). Part 3: Using asymmetric techniques (IS 1997).

[15] ISO/IEC 14888:1998 Information technology – Security techniques – Digital Signatures with appendix.

[16] Lai, Xuejia; Massey, James: A proposal for a New Block Encryption Standard (IDEA). Advances in Cryptology – Eurocrypt'90 Proceedings, Berlin: Springer 1991, pp.389-404.

[17] National Bureau of Standards: Data Encryption Standard (DES). FIPS PUB 46-1, Jan 1988.

[18] National Bureau of Standards: Secure Hash Standard (SHS-1). FIPS PUB 180-1, 17.4.1995.

[19] National Computer Security Center: Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria. NCSC-TG-021, Version 1, Apr 1991.

[20] National Computer Security Center: Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (Red Book). NCSC-TG-005, Version 1, Jul 1987.

[21] National Institute of Standards and Technology: Digital Signature Standard (DSS). NIST FIPS PUB 186, May 1994.

[22] NATO: NATO Trusted Computer System Evaluation Criteria (Blue Book). NATO AC/35-D/1027, 1987.

[23] Rechenberger, Peter; Pomberger, Gustav: Informatik-Handbuch. München / Wien: Carl Hanser 1999.

[24] Rivest, Ronald L.: The MD5 Message Digest Algorithm. RFC 1321, Apr 1992.

[25] Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard A.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, Vol.21, Nr.2, Feb 1978, pp.120-126.

[26] Schneier, Bruce: Applied Cryptography. John Wiley & Sons, Inc. 1996, p. 469.

[27] Tanenbaum, Andrew S.: Verteilte Betriebssysteme. München: Prentice Hall 1995.

12 Biographie

Petra Wohlmacher studierte Mathematik mit Schwerpunkt Informatik an der Technischen Universität Darmstadt. Von 1994 bis 1997 war sie als wissenschaftliche Mitarbeiterin im Forschungsbereich SmartCardTechnologie am Institut für Telekooperationstechnik beim GMD – Forschungszentrum für Informationstechnik GmbH in Darmstadt beschäftigt. Seit dem 1.7.1997 arbeitet sie als Universitätsassistentin an der Universität Klagenfurt am Lehrstuhl für Systemsicherheit. Ihre Forschungsschwerpunkte sind Kryptographie, Sicherheitsinfrastrukturen, Chipkartentechnologie und innovative Smartcard-Anwendungen. Seit dem 1.1.1997 ist sie Mitglied der GI-Fachgruppe 2.5.3 "Verläßliche IT-Systeme" (VIS), deren Leitungsgremium sie seit dem 1.10.1997 angehört.