# Sicherheitsmechanismen I

Prof. Gunnar TEEGE
Frank EYERMANN
Matthias GÖHNER
Christian OPINCARU
(Hrsg.)

Institut für Informationstechnische Systeme, IIS

Bericht Nr. 2005-07
Juli 2005

Universität der Bundeswehr München

Fakultät für

**INFORMATIK**

Werner-Heisenberg-Weg 39 • D-85577 Neubiberg

# Inhalt

Sicherheitsaspekte in der Informationstechnik spielen aktuell eine große Rolle. Die Entwicklungen betreffen nicht nur die Informatik-Techniker, sondern haben in zunehmendem Umfang Auswirkungen auf das tägliche Leben. Aus diesem Grund wurde im Frühjahr 2005 erstmalig ein Seminar zu Sicherheitsmechanismen am Institut für Informationstechnische Systeme IIS der Universität der Bundeswehr München durchgeführt. Es ergänzt und vertieft das Thema, das bereits in mehreren Lehrveranstaltungen im Rahmen des Informatik-Studiums behandelt wird.

Der inhaltliche Schwerpunkt des Seminars lag auf der Anwendung existierender und in der Entwicklung befindlicher Sicherheitsmechanismen. Entsprechend breit waren die Themen der einzelnen Vorträge verteilt. Nach einer Einführung, die die wesentlichen Grundlagen für die Teilnehmer rekapituliert, befassen sich die nächsten zwei Vortrage mit der "klassischen" Anwendung zum Zugangsschutz zu Rechnern. Die Vorträge betrachten die Nutzerauthentifizierung unter den Betriebssystemen Windows und Unix.

Es folgen zwei Vorträge zu Anforderungen und Lösungen von Sicherheitsmechanismen im Zusammenhang mit Web Services. Diese aktuelle Technik zum Aufbau verteilter Systeme auf der Basis des WWW hat durch ihren Einsatz in allen Bereichen des täglichen Lebens teilweise sehr hohe Anforderungen an Sicherheit. Es ist zu erwarten, dass in naher Zukunft dieses Thema eine wichtige Rolle spielen wird.

Während Web Services in erster Linie die Ebene der Anwendungen betrifft, behandeln die nächsten beiden Vorträge Sicherheitsaspekte auf der Ebene der Kommunikationsinfrastruktur. Systeme zu Intrusion Detection und Intrusion Prevention sind heute eine wichtige Voraussetzung zum Betrieb sicherer Kommunikationsnetze, das neue Internet-Protokoll IPV6 berücksichtigt bereits gewisse Sicherheitsanforderungen.

Der nächste Vortrag betrachtet das Gebiet des digitalen Rechtemanagements DRM, das zunehmend für Verbraucher im Bereich elektronischer Medien eine Rolle spielt. Ähnlich relevant wird für Verbraucher in naher Zukunft das Thema "Biometrie" werden, das in den abschließenden beiden Vorträgen vorgestellt wird.

Die Teilnehmer am Seminar erhielten zu Beginn eine Einführung in Methoden der wissenschaftlichen Literaturrecherche im Internet. Auf dieser Basis konnten sie sich eigenständig relevante Informationen zum jeweiligen Thema erarbeiten. Engagierte Unterstützung dabei und bei der Durchführung des Seminars leisteten die Mitarbeiter des IIS Matthias Göhner, Frank Eyermann und Cristian Opincaru. Mein Dank gilt allen Beteiligten, die tatkräftig zum Gelingen des Seminars beigetragen haben.

Gunnar Teege
Juli 2005

# Inhaltsverzeichnis

# Kapitel 1

# Allgemeine Grundlagen der IT-Sicherheit

*Alexander Talaska*

*Dieses Kapitel soll eine Einführung in die Grundlagen der IT-Sicherheit sein. Es wird erläutert, warum Sicherheit in der Informationstechnik heute so wichtig ist, welche Ziele sie verfolgt und gibt einen Einblick welche Bedrohungen sie gefährden. Außerdem soll hier ein Überblick gegeben werden, mit welchen logischen, organisatorischen, sowie technischen Maßnahmen dies erreicht werden soll.*

## Inhaltsverzeichnis

## 1.1 Was ist IT-Sicherheit?

Die Motivation für die Wichtigkeit von IT-Sicherheit ist für jeden Benutzer eines Computers leicht zu geben, insbesondere wenn dieser Teil eines Netzwerkes ist. Fast jeder Rechner in unseren Breiten ist heutzutage mit dem Internet verbunden und ihre Benutzer haben wahrscheinlich schon einige Erfahrung mit Viren gesammelt. Doch IT-Sicherheit geht noch viel weiter. Insbesondere kommerzielle Anwendungen wie zum Beispiel E-Banking oder Datenbankdienste für Versicherungen oder Ämter zeigen weitere Aspekte dieses Themenbereichs auf. Niemand möchte, dass Unbefugte Zugriff auf seine persönlichen Daten oder gar das eigene Bankkonto haben. Spätestens, wenn es um Anwendungen geht, die die öffentliche Sicherheit gefährden könnten, wird klar, dass es notwendig ist Maßnahmen zu ergreifen, die Missbrauch oder Angriffe ausschließen.

Wie dieser kurzen Motivation bereits zu entnehmen ist soll es im Folgenden hauptsächlich um den Teilbereich der IT-Sicherheit gehen, der im Englischen mit dem Wort *security* beschrieben würde und sich mit dem Schutz vor böswilligen Angriffen oder unbefugtem Zugriff auf IT-Systeme beschäftigt. Es sollen die Ziele und einigen Maßnahmen zu ihrer Erreichung beleuchtet werden. Zuerst jedoch ein paar grundlegende Informationen.

### 1.1.1 Ziele und Bedrohungen der IT-Sicherheit

#### Ziele der IT-Sicherheit

Um Sicherheit in IT-Systemen zu gewährleisten müssen im großen und ganzen drei Hauptziele erreicht werden. Diese sind **Vertraulichkeit**, **Integrität** und **Verfügbarkeit**. Anwendungsspezifisch werden sie ergänzt durch die Forderung nach **Authentizität**, **Verbindlichkeit** und **Anonymität**.

- **Vertraulichkeit** soll verhindern, dass Informationen durch Unbefugte erlangt werden können oder zugänglich gemacht werden. Nur wenn ausschließlich berechtigte Personen oder Personengruppen auf eine Information zugreifen können kann diese als vertraulich bezeichnet werden.
  *Beispiel: Niemand außer dem Kontoinhaber und seiner Bank sollen seinen Kontostand kennen.*

- **Integrität** soll garantieren, dass nur erlaubte und beabsichtigte Änderungen der Informationen in einem IT-System stattfinden.
  *Beispiel: Überweisungen vom eigenen Konto darf nur der Kontoinhaber in Auftrag geben.*

- **Verfügbarkeit** soll gewährleisten, dass die Benutzer eines IT-Systems einen Dienst in angemessener Qualität, Form und Zeit nutzen können.
  *Beispiel: E-Mail Server sollen jederzeit in der Lage sein Nachrichten anzunehmen und zeitnah an den Adressaten weiterzuleiten.*

- **Authentizität** kann in zwei Unterbereiche gegliedert werden. Zum Ersten die Datenauthentizität, die gewährleistet, dass die Daten die zwischen oder in Systemen genutzt werden echt und nicht etwa unbefugte Kopie sind. Der zweite Aspekt der Authentizität bezieht sich auf die Parteien, die an einer Datenkommunikation teilnehmen. Auch diese sollen echt und diejenigen, als die sie sich ausgeben, sein.
  *Beispiel: Sowohl der Inhalt als auch der Name des Absenders einer E-Mail sollen echt sein.*

- **Verbindlichkeit** bedeutet, dass Sicherheit besteht, ob geforderte Operationen ausgeführt wurden.
  *Beispiel: Nach dem versenden einer E-Mail muss der Absender vom Erfolg des Vorgangs ausgehen können und andernfalls benachrichtigt werden.*

- **Anonymität** heißt, dass die eigene Identität und Daten nur soweit wie nötig oder gewollt preisgegeben werden.
  *Beispiel: Beim Besuch einer Seite im Internet soll der Inhaber meine Daten nur erhalten wenn ich sie im übermitteln möchte.*

### Bedrohungen für die IT-Sicherheit

Um Informationen wirksam zu schützen muss zuerst eine Analyse der möglichen Bedrohungen durchgeführt werden. Diese können unterschieden werden nach Bedrohungsziel, -ort und -klasse [1].
Die Bedrohungsziele sind die eben genannten Ziele der IT-Sicherheit.
Als Bedrohungsorte kommen zwei Komponenten in Frage zum ersten der Computer, auf dem die Daten verarbeitet oder gespeichert werden, und zum zweiten der Kommunikationsweg, auf dem die Informationen übertragen werden. Demzufolge unterscheidet man hier zwischen Computer- und Kommunikationssicherheit.
Bedrohungsklassen beschreiben die eigentliche Ursache möglicher Schäden. Sie werden unterteilt in [1]:

- **Höhere Gewalt:** Störung im operativen Betrieb oder Ausfall aufgrund von natürlichen Ereignissen
- **Organisatorische Mängel:**zum Beispiel unklare oder unbekannte Zuständigkeiten bzw. Verantwortlichkeiten
- **Bedienungsfehler:** treten in vielfältiger Weise auf, z.B. falsche Eingaben
- **Technisches Versagen:** Ausfall oder Störung von technischen Komponenten des Systems
- **Vorsätzliche Handlungen:** große und wachsende Bedrohung durch bösartige Angriffe auf ein System

Daraus läßt sich nun ein dreidimensionales Bedrohungsmodell ableiten (siehe Abb.1.1).

Um ein System nun wirkungsvoll zu schützen müssen alle Felder dieses Bedrohungsraumes durch entsprechende Maßnahmen abgedeckt werden.

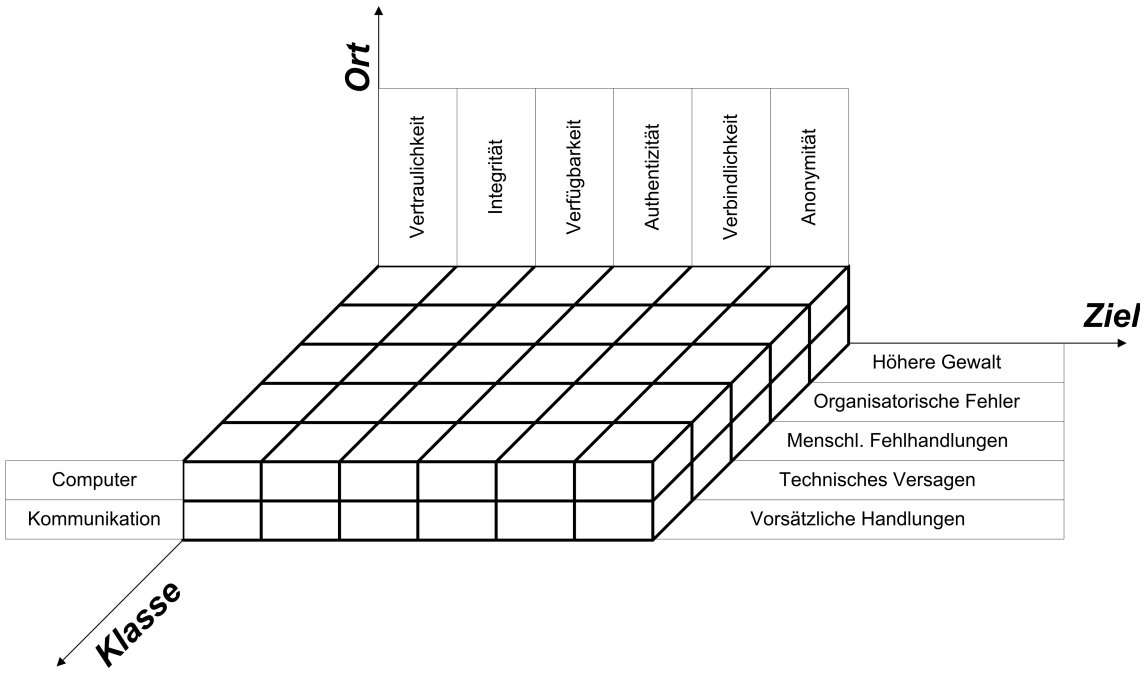


Abbildung 1.1: Erweitertes Bedrohungsmodell nach [1]

Als Beispiel kann die Vertraulichkeit auf dem Computer durch Zugriffsbeschränkungen und -kontrollen vor vorsätzlichen Angriffen geschützt werden. Auf dem Kommunikationsweg können solche Angriffe durch Verschlüsselung der Informationen vereitelt werden.

Obwohl solche Sicherheitsmaßnahmen oft konkret zur Erreichung eines dieser Ziele dienen ist die Erreichung dieser dennoch teilweise eng miteinander verknüpft. Angriffe auf die Verfügbarkeit eines Systems sind zum Beispiel am leichtesten und wirkungsvollsten aus dem System selbst heraus durchzuführen. Vorkehrungungen, die die Authentizität und Integrität sicherstellen, tragen also automatisch auch zur Erhaltung der Verfügbarkeit bei. Andererseits müssen jedoch auch stets Kompromisse zwischen den Maßnahmen gefunden werden. Ein System, das durch aufwendige Zugangskontrollen geschützt und von der Außenwelt getrennt ist, bietet offensichtlich ein hohes Maß an Authentizität und Vertraulichkeit, hat aber sehr schlechte Eigenschaften bezüglich der Verfügbarkeit. Die geforderte Funktionalität, Anforderungen an den Bedienkomfort sowie finanzieller, materieller und zeitlicher Aufwand schränken die Möglichkeiten ein System abzusichern weiterhin stark ein.

### 1.1.2 Generische Angriffe

Als generische Angriffe bezeichnet man solche Angriffsmuster, die besonders häufig ausgeführt werden und zum Erfolg führen. Diese Angriffe können verschiedenste Formen haben und oft helfen nur organisatorische Maßnahmen, um sich vor ihnen zu schützen. Einige Beispiel sollen hier erwähnt werden.

Als **Trojanische Pferde** bezeichnet man scheinbar harmlose Programme, die schädliche Teile enthalten. Dies kann zum Beispiel ein Spiel sein, das im Hintergrund Daten löscht.

Ein Vorteil dieser Methode liegt darin, dass der Angreifer nicht selbst ins System eindringen muss. Den Ausführungsbefehl für die schädliche Datei gibt nämlich der Benutzer selbst.

**Pufferüberläufe** treten auf, wenn Programme auf Speicher zugreifen, der nicht für diese reserviert ist dadurch können Daten verloren gehen oder durch die Veränderung von Rücksprungadressen schädliche Programme zur Ausführung kommen.

Aber auch die Entwickler und Betreuer eines Systems können Gefahrenquellen darstellen. Beispiele hierfür sind **Logische Bomben** und **Versteckte Hintertüren**.
Logische Bomben sind schädliche Programmteile, die in eine Software eingebaut werden und solange nicht zur Ausführung kommen bis der Angreifer dies will. Dieser Zeitpunkt könnte zum Beispiel die Entlassung aus dem Unternehmen sein. Versteckte Hintertüren werden durch Programmierer eingebaut und bieten die Möglichkeit spätere Zugangskontrollen, zum Beispiel einer Bankensoftware, zu umgehen.

Ein weiterer Ansatz besteht darin die Benutzer eines Systems zu täuschen, um Zugangsdaten oder andere Informationen zu erhalten. Dies kann durch manipulierte oder gefälschte Login-Masken (**Login Spoofing**) geschehen oder durch gefälschte E-Mails, die den Empfänger dazu bringen sollen seine Daten weiterzusenden (**Phishing**).

Diese Methode bildet auch einen Grenzfall zu **Social Engineering**, welches nichttechnische Angriffe direkt auf die Benutzer eines Systems umfasst. Ein häufiges Beispiel hierfür ist der Telefonanruf um ein „vergessenes “ Passwort zu erfragen.

#### Organisatorische Schutzmaßnahmen

Um IT-Systeme vor solchen Angriffen ist es neben entsprechenden technischen Maßnahmen auch auf organisatorischer Ebene vorzusorgen.

Wichtig ist es vor und im Einsatz der Systeme durch gezielte Angriffe auf häufige Schwachstellen mögliche Gefährdungen zu erkennen und auszuschalten. Eine Möglichkeit Bedrohungen von Seiten der Entwickler und Administratoren auszuschließen stellen Code Reviews und gegenseitige Kontrolle dar. Rechte sollten so verteilt werden, dass niemand unbemerkt das System manipulieren kann. Regelmäßige Belehrungen der Mitarbeiter begleitet von Kontrollen können außerdem die Gefahr, die vom Social Engineering ausgeht, reduzieren.

## 1.2 Grundlegende Mechanismen für die IT-Sicherheit

In diesem Abschnitt sollen einige Grundlegende Sicherheitsmechanismen dargestellt und erklärt werden. Hauptziel der folgenden Ansätze sind vor allem der Schutz von Integrität und Authentizität.

### 1.2.1 Authentifikation und Autorisierung

#### Zugangskontrolle

Wie bereits erwähnt bedeutet Vertraulichkeit, dass nur berechtigte Personen oder Personengruppen Zugang zu einem IT-System haben und auf die enthaltenen Informationen zugreifen können. Dazu muss sich die jeweilige Person beweisen, dass sie zu eben diesen gehört. Hierfür gibt es verschiedene Prinzipien. Ein Benutzer kann seine Identität belegen indem er etwas **weiß** (z.B. ein Passwort), **besitzt** (z.B. eine Chipkarte) oder **ist** (durch ein biometrisches Merkmal)[2].

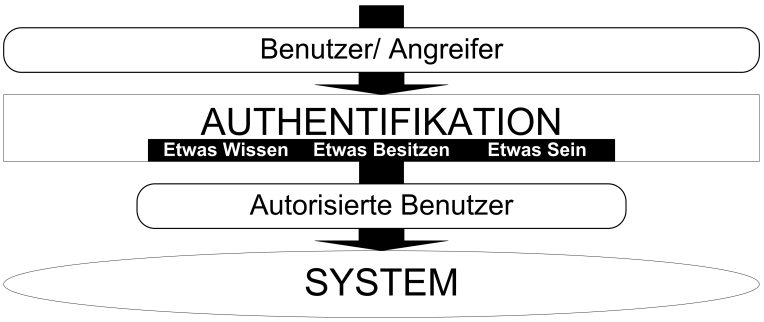


Abbildung 1.2: Nur berechtigte Benutzer erhalten Zugang zum System

Die Authentifikation durch Passwörter ist die am häufigsten angewandte Methode, die Identität eines Benutzers zu verifizieren, denn diese Methode ist leicht zu verstehen und zu implementieren. Es werden Tupel von Benutzernamen und ihrem Passwörtern gespeichert. Bevor der zu schützende Bereich betreten wird, muss der Anwender die jeweiligen Daten angeben.
Ein Nachteil dieser Methode ist, das viele Benutzer sehr einfache und häufig vorkommende Worte und Verhaltensmuster bei der Auswahl ihrer Passworte wählen. Der eigene Name, Geburtsjahr oder Ähnliches können leicht erraten werden und zum Eindringen in ein System genutzt werden. Werden sicherere Zugangsdaten, die nicht erraten werden können vorgegeben, hat dies den Nachteil, dass sich der Benutzer nur schwer merken kann, was den Komfort deutlich einschränkt.
Eine Erweiterung dieses Systems stellen die sogenannten Einmal-Passwörter dar, die wie der Name beschreibt nach einmaliger Benutzung ihre Gültigkeit verlieren. Diese Form der Zugangskontrolle ist natürlich komplexer als die vorhergehende, da der Benutzer die nachfolgenden Passwörter kennen muss, bietet aber eine höhere Sicherheit.
Im Bankgeschäft findet sich diese Methoden beim Online-Banking wieder. Der Kunde verschafft sich Zutritt zum System, indem er seine Kontonummer und ein Passwort eingibt. Danach kann er auf sein persönlichen Daten, wie Kontostand und Umsätze zugreifen. Um jedoch empfindlichere Anweisungen wie das Einrichten eines Dauerauftrages oder das Veranlassen einer Überweisung durchzuführen muss er diese durch eine 6-stellige TAN (Transaktionsnummer) bestätigen. Diese ist ein Einmal-Passwort und der Kunde entnimmt sie einer Liste, die er von der Bank erhält und die nur er kennt.

Eine andere Möglichkeit die Identität des Benutzers zu überprüfen, besteht darin, ihm eine Frage/Herausforderung (*challenge*) zu stellen, die nur er beantworten/erfüllen kann

(*response*). Die Möglichkeiten der sogenannten **Challenge-Response Verfahren** sind sehr vielfältig und können beliebig komplex gestaltet werden. Ein einfaches Verfahren wäre die Frage nach persönlichen oder anderen vorher vereinbarten Angaben (z.B. *„Wie lautet der Mädchenname ihrer Mutter?“*, *„Wie heißt ihr Haustier?“*). Komplexer und damit auch sicherer sind Challenges die sich auf vergangene Transaktionen beziehen oder deren Antwort sich nur mittels (ausschließlich den beteiligten Parteien bekannten) Algorithmen oder Schlüsseln errechnen lässt.

Der Ansatz etwas zu besitzen wird realisiert mit der **Authentifizierung durch Gegenstände**.
Diese Form der Zugangskontrolle ist schon seit Jahrhunderten bekannt und erfolgt zum Beispiel durch einen einfachen Schlüssel aus Metall, der benötigt wird, um eine Tür zu öffnen. Heutzutage sind Chipkarten ein weit verbreitetes Mittel, um sich Zugang zu bestimmten Einrichtungen, Daten oder Dienstleistungen zu verschaffen. Solche Karten sind mit einem Magnetstreifen oder einem Chip versehen, auf denen entsprechende Zugangsinformationen gespeichert sind.
Da zum Beispiel Magnetkarten, die auch als Konto- bzw. Kreditkarten eingesetzt werden, inzwischen sehr leicht zu lesen und zu kopieren sind, wird meist ein zusätzlicher Nachweis, wie die Eingabe eines Passwortes (PIN) oder eine Unterschrift gefordert.
Leistungfähigere Nachfolger dieser Generation von Karten werden und sind teilweise die sogenannten *Smart Cards*. Sie enthalten im Gegensatz zu den üblichen Chipkarten auch einen kleinen Prozessor und sind somit in der Lage, das bereits erwähnte Challenge-Response Verfahren zu nutzen, also zum Beispiel auf eine Challenge wie eine Zeichenkette eine passende Antwort zu errechnen.

Auf die Authentifikation durch **Biometrische Merkmale** wird in einem späteren Kapitel näher eingegangen. Die Grundidee besteht darin, das einige Merkmale bei jedem Menschen einzigartig sind und somit zum Nachweis seiner Identität dienen können. Beispiele für diese Merkmale sind der Fingerabdruck oder die Netzhaut.

### Autorisierung

Um den Zugang zu IT-Systemen und Informationen durch Authentifikationsverfahren zu schützen muss nun natürlich festgelegt werden, wer Zugang zu diesen erhält und welche Rechte derjenige hat. Auch hier gibt es verschiedene Strategien.
Die Rechte können sowohl bei den Ressourcen bzw. Informationen (z.B. in *AC-Listen*) oder beim Benutzer (z.B. in *C-Listen*) gespeichert werden.

In ACLs (Access-Control-List) wird gespeichert, welche Benutzer oder Domänen welche Rechte besitzen. Sie haben den Vorteil, dass der Besitzer einer Ressource, zum Beispiel einer Datei, die Zugriffsrechte sehr leicht verwalten kann. Sollen jedoch einem Benutzer alle Rechte entzogen werden ist dies mit sehr großem Aufwand verbunden, da alle Ressourcen des Systems durchsucht werden müssen.

Durch die Verteilung von Capabilities, also den C-Listen, werden die Rechte beim Benutzer gespeichert. Sie sind Schlüssel, die bei jeder Anfrage an eine Ressource mitgesendet werden müssen. Dies bietet den Vorteil, dass ein Benutzer seine Rechte sehr leicht an andere Benutzer oder seine Prozesse weitergeben kann. Ein gravierender Nachteil ist jedoch, dass das selektive entziehen von Rechten sehr schwierig ist, denn es muss ein neuer Schlüssel

erstellt werden und an allen Benutzern mitgeteilt werden, die weiterhin Zugriffsrechte behalten sollen. Dazu müssen natürlich alle Benutzer, die einen Schlüssel erhalten sollen bekannt sein.

### 1.2.2 Kryptosysteme

#### Grundlagen der Kryptographie

Um zu verhindern, dass unbefugte Personen während der Übertragung von Daten auf Informationen zugreifen oder diese ungewollt verändern gibt es zwei mögliche Ansätze.
Der erste Ansatz, die so genannte Steganographie, besteht darin, Informationen oder ihre Übertragung zu verbergen. Dieser Ansatz wurde früher sehr oft verwandt zum Beispiel in Form von Geheimtinte oder Mikrofilmen. Heutzutage gewinnt jedoch die zweite Möglichkeit ein deutliches Übergewicht. Diese besteht darin die Existenz und die Übertragung von Informationen nicht zu verbergen, sondern durch Verschlüsselung unlesbar zu machen und durch weitere Mechanismen so zu bearbeiten, dass eine ungewollte Manipulation erkannt wird.
Diesen Bereich nennt man Kryptologie. Sie setzt sich im wesentlichen zusammen aus der Kryptographie, der Erzeugung von Chiffretext aus Klartext mittels eines Schlüssels und umgekehrt, und der Kryptoanalyse, dem Entschlüsseln von Chiffretext, um den verwendeten Schlüssel zu ermitteln. Schwerpunkt hier soll das Gebiet der Kryptographie sein.

Grundidee der Kryptographie ist es, dass ein für jeden lesbarer Text (*Klartext* oder $P$) in eine Darstellung umgeformt wird, die nur von den Personen gelesen werden kann, die die Verschlüsselungsinformation (üblicherweise wird diese $E$ genannt) kennen. Diese Darstellungsform wird Chiffretext ($C$) genannt. Hierbei haben Klar- und Chiffretext je ein eigenes Alphabet, welche unterschiedlich groß sein können.
Eine einfache Verschlüsselung ist also schon die Umformung von Dezimal in Binär oder Hexadezimalzahlen. Auch hier sind die Informationen, also der Wert der Dezimalzahl, nicht ohne weiteres zu erkennen und die Alphabete sogar unterschiedlich groß.

Ein Chiffretext kann erneut Eingabe für einen Schlüssel sein. Eine solche Mehrfachverschlüsselung entsteht aus der Komposition von Verschlüsselungsinformationen.
Man unterscheidet hier zwischen Produktchiffrierung, bei der die verwendeten Schlüssel nicht notwendigerweise statistisch unabhängig sind, und Kaskadenchiffrierung, bei der die Schlüssel statistisch unabhängig sind. Bezüglich der Sicherheit von Produktchiffrierung können keine allgemein gültigen aussagen getroffen werden. Bei Kaskadenchiffrierung gilt, dass die Verschlüsselung mindestens so sicher ist wie ihre Komponenten.

Die der Verschlüsselung entsprechende Entschlüsselungsinformation wird $D$ genannt und ist die Umkehrfunktion der Verschlüsselung. Allerdings existiert diese Entschlüsselungsinformation nicht immer.

Da die Verwendung von immer gleichen Verschlüsselungs- und Entschlüsselungsinformationen nicht besonders sicher und die häufige Entwicklung von neuen sehr aufwendig und unkomfortabel ist wurde der Einsatz von sogenannten Schlüsselwerten (*keys*) eingeführt. Zusätzlich zum Klartext hängt das Ergebnis der Verschlüsselung vom Wert des Schlüssels

ab. Dies hat den Vorteil, dass viele Parteien die gleiche Verschlüsselungsinformation verwenden können und Informationen trotzdem sicher vor unbefugtem Zugriff sind, solange nur berechtigte Personen den Schlüsselwert kennen.

Dieses Prinzip wird *Das Prinzip von Kerkhoffs* genannt. Es besagt, dass die Sicherheit nicht von der Geheimhaltung der Ver- und Entschlüsselungsinformationen abzuhängen hat, sondern ausschließlich von der Geheimhaltung der verwendeten Schlüssel ($K$).

Weitere wichtige Prinzipien für den Entwurf von Kryptosystemen wurden durch Claude E. Shannon vorgeschlagen [1].
Eines ist das Prinzip der *Konfusion*, welches empfiehlt, dass der funktionale Zusammenhang zwischen Klartext, Chiffretext und Schlüssel möglichst komplex sein sollte. Das zweite ist die *Diffusion*. Sie gibt vor, dass jedes Zeichen im Chiffrecode von möglichst vielen Zeichen des Klartextes und dem gesamten Schlüssel abhängen soll, um statistische Besonderheiten und Eigenschaften des Textes zu verschleiern. Zum Beispiel das häufige Auftreten von Zahlen oder aber häufig benutzte Buchstaben einer Sprache, die durch sich wiederholende Muster im Chiffretext erkannt werden könnten. Diese Regeln gelten insbesondere für Verschlüsselungen, bei denen der Klartext deutlich länger als der Schlüssel ist.

Um die Güte eines Kryptosystems zu beurteilen hat Shannon desweiteren diese fünf Kriterien formuliert [1].

1. *„Amount of secrecy“*

   Man unterscheidet verschiedene Grade der Sicherheit. Berechenbar sicher heißt, dass selbst mit den bestmöglichen Angriffen das Analysieren des Code unmöglich aufwändig wäre. Praktisch sicher bedeutet, dass der beste bekannte Angriff auf ein System nicht einfacher ist als die vollständige Schlüsselsuche, also das Ausprobieren jeden möglichen Schlüssels.

2. *„Size of Key“*

   Da Schlüssel auch gespeichert und übertragen werden müssen, dürfen sie nicht zu groß sein. Andererseits, ist die Länge des Schlüssels ein wichtiger Faktor für die Sicherheit der Verschlüsselung. Ist der Schlüssel zu kurz kann er durch Angreifer leichter „geknackt“ werden.

3. *„Complexity of Enciphering and Deciphering Operations“*

   Dieses Kriterium hatte ursprünglich die Absicht Operationen nicht zu komplex zu gestalten, um Fehler bei manueller Ausführung und zu teure Automaten zu vermeiden. Heutzutage werden Verschlüsselungen allerdings kaum noch per Hand ausgeführt werden und moderne Rechner können beinahe beliebig komplexe Berechnungen ausführen. Allerdings können sich Angreifer diese Eigenschaft ebenfalls zunutze machen. Deshalb muss zwischen Komplexität (Sicherheit) und Geschwindigkeit (Komfort/ Performance) ein Kompromiss gefunden werden.

4. *„Propagation of Errors“*

   Fehler sollten sich nur hinreichend gering ausbreiten. Das bedeutet, dass die negativen Folgen eines Fehlers im System oder eines erfolgreichen Angriffs begrenzt

sind und dass die Konsequenzen ein gefordertes Mindestmaß an Sicherheit nicht gefährden.

5. *„Expansion of Messages“*

   In der Regel sollte die Größe des Chiffretextes die des Klartextes nicht oder nur unwesentlich überschreiten. Dieses Kriterium spielt in erster Linie in Hinsicht auf die Leistungsfähigkeit eines Systems eine wichtige Rolle.

### Symmetrische und asymmetrische Verschlüsselung

Beim Entwurf von Verschlüsselungs- und Entschlüsselungsverfahren gibt es zwei Alternativen. Wird in beiden Schritten der gleiche Schlüsselwert verwendet spricht man von symmetrischer ansonsten von asymmetrischer Verschlüsselung (Abb.1.3). Beide Möglichkeiten haben Vor- und Nachteile.

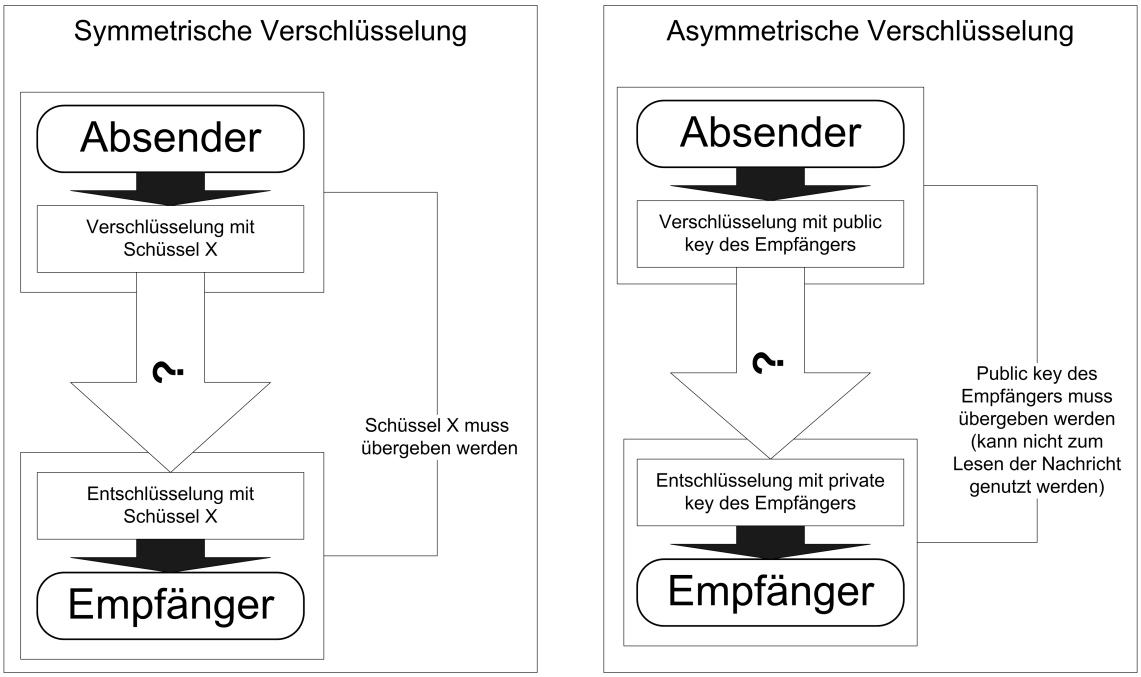


Abbildung 1.3: Ablauf von symmetrischer und asymmetrischer Verschlüsselung

Symmetrische Verschlüsselung (z.B. DES, Triple DES) funktioniert relativ schnell hat allerdings auch Nachteile. Der Schlüssel muss übertragen werden, damit der Empfänger die Nachricht entschlüsseln kann. Ist dieser nun auch einem Angreifer bekannt, kann dieser die Informationen lesen und sogar verändert weiter senden. Außerdem muss ein sehr großer Aufwand bei der Schlüsselverwaltung betrieben werden. Für jeden Kommunikationspartner , evtl. sogar jede Nachricht, muss ein neuer Schlüssel generiert und unter Umständen gespeichert bzw. übertragen werden.

Bei der asymmetrischen Verschlüsselung (z.B. RSA) liegt der Ansatz darin, dass jeder Kommunikationsteilnehmer einen privaten, also nur ihm bekannten Schlüssel, und einen öffentlichen Schlüssel besitzt. Eine mit dem privaten Schlüssel eines Teilnehmers chiffrierte Nachricht kann nur mit dem entsprechenden öffentlichen Schlüssel lesbar gemacht werden

oder umgekehrt. Dies reduziert den Aufwand bei der Schlüsselverwaltung enorm, da nur der eigene private Schlüssel und die öffentlichen Schlüssel der Kommunikationspartner bekannt sein müssen. Der Nachteil hier liegt darin, dass asymmetrische Verschlüsselung relativ langsam ist, was insbesondere bei leistungsschwächeren Rechnern oder großen Datenmengen problematisch werden kann.

Durch die geschickte Kombination beider Verfahren können jedoch die jeweiligen Vorteile genutzt und deren Nachteile nahezu ausgeschaltet werden. Der *Diffie-Hellman-Algorithmus* [4] setzt genau dies um, indem man die eigentliche Nachricht mit einem Einmalschlüssel symmetrisch verschlüsselt und diesen Schlüssel asymmetrisch verschlüsselt mitsendet. Der Empfänger ist nun in der Lage den symmetrischen Schlüssel mit seinem privaten Schlüssel zu entziffern und damit die Nachricht schnell lesbar zu machen. So können Nachrichten schnell chiffriert bzw. dechiffriert werden und trotzdem die Vorteile der leichteren Schlüsselverwaltung genutzt werden.

Die größte Gefahr für Authentizität und Integrität besteht nun darin, dass ein Angreifer Nachrichten abfängt und möglicherweise sogar verändert an weiterleitet (*Man-In-The-Middle-Angriff*, siehe Abb.1.4).

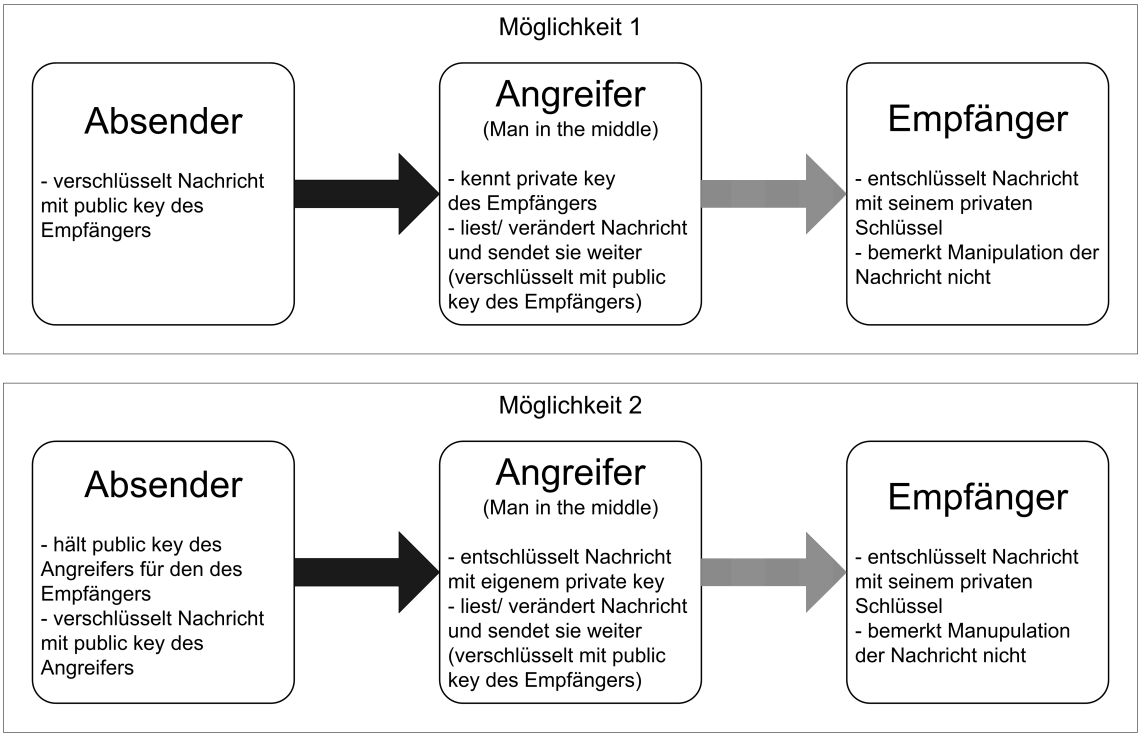


Abbildung 1.4: Man-In-The-Middle-Angriff

In Möglichkeit 1 ist ein Fall dargestellt, in dem der Angreifer den privaten Schlüssel des Empfängers in Erfahrung gebracht hat. Wird nun eine Nachricht an diesen gesandt und es gelingt dem Angreifer die Nachricht abzufangen, kann er die Nachricht mit dem privaten Schlüssel des Empfängers dechiffrieren. Nachdem er sie gelesen und eventuell manipuliert hat, verschlüsselt die Informationen wiederum mit dem öffentlichen Schlüssel von $E$ und sendet sie weiter. Der Empfänger der Nachricht erhält nun die veränderte Nachricht.

Um diesen Angriff zu vereiteln müsste die Nachricht so markiert werden, dass eine mögliche Änderung erkannt wird. Diese Markierung darf demzufolge nur durch den Absender zu erstellen sein.

Die zweite Möglichkeit zeigt, wie der Absender getäuscht wird. Schon bei der Übergabe des öffentlichen Schlüssels von $E$ an $A$ muss muss es dem Angreifer gelingen diesen abzufangen und gegenüber dem Absender den eigenen public key als den des Empfängers auszugeben. Die später mit diesem Schlüssel chiffrierten Nachrichten von $A$ sind nun für den Angreifer lesbar. Fängt er diese ab, kann er sie ebenfalls unbemerkt verändern und weitersenden.
Eine Markierung des Absenders kann in diesem Fall zwar helfen, nützt jedoch nichts, falls auch $E$ den falschen öffentlichen Schlüssel erhält, da der Angreifer dann eine eigene Markierung erstellen kann (Siehe *Digitale Unterschriften*). Hier muss also ein Weg gefunden werden, die öffentlichen Schlüssel so zu verbreiten, dass diese für die Beteiligten vertrauenswürdig (authentisch) sind.

Um Gewissheit zu erlangen, dass die Nachrichten wirklich vom angegebenen Absender stammen und nicht verändert wurden, werden die folgenden Verfahren genutzt.

### 1.2.3 Digitale Unterschriften

Digitale Unterschriften oder **Signaturen** geben dem Empfänger einer Nachricht die Möglichkeit durch Prüfsummen (*Hashes*) festzustellen, ob diese auf dem Übertragungsweg manipuliert worden ist.
Sie sind das Ergebnis der Eingabe der gesamten Nachricht in eine Hash-Funktion, die die Eigenschaft hat, dass schon bei geringfügiger Änderung des Eingabewertes ein völlig anderes Ergebnis errechnet würde. Durch Verschlüsselung des Hashwertes mit dem privaten Schlüssel des Absenders entsteht die eigentliche Signatur, die an die Nachricht angehängt wird.
Nach Erhalt der Nachricht kann der Empfänger die Signatur mittels des öffentlichen Schlüssels des Absenders entschlüsseln und außerdem selbst die Prüfsumme für die erhaltene Nachricht errechnen. Falls beide Werte nicht übereinstimmen oder das Entschlüsseln fehlschlägt kann davon ausgegangen werden, dass die Nachricht verändert wurde oder nicht vom angeblichen Absender stammt.

### 1.2.4 Zertifikate

Dem Angreifer bleibt nun aber noch die Möglichkeit den Empfänger einer Nachricht zu täuschen, indem er seinen öffentlichen Schlüssel als den des Absenders ausgibt. Er könnte die Nachricht lesen und sogar verändern und mit einer neuen Signatur versehen weitersenden.
Um Nachrichten und ihre Signaturen nun absolut glaubwürdig zu machen, muss sich der Empfänger vergewissern, dass der öffentliche Schlüssel wirklich zum eigentlichen Absender gehört. Dies kann durch *Zertifikate* geschehen. Sie dienen der sicheren Schlüsselverteilung und enthalten neben den persönlichen Daten des Inhabers (der Absender der Nachrichten)

und seinem öffentlichen Schlüssel auch den Namen der Stelle, die das Zertifikat ausgestellt hat (*Zertifizierungsstelle*), eine Seriennummer, Angaben zur Gültigkeitsdauer und wiederum eine Signatur, die mit dem privaten Schlüssel der Zertifizierungsstelle erstellt wird.

Um das Prinzip dieses Systems zu veranschaulichen, kann ein einfaches Szenario skizziert werden (siehe auch Abb.1.5).
Die Akteure sind der Empfänger ($E$), der Absender ($A$) und eine Zertifizierungsstelle ($Z$). Sowohl $A$ als auch $Z$ verfügen über einen eigenen privaten und einen öffentlichen Schlüssel ($A_{priv}$, $A_{pub}$, $Z_{priv}$, $Z_{pub}$).
Weiterhin nehmen wir an, dass der Empfänger der Zertifizierungsstelle $Z$ vertraut und ihren öffentlichen Schlüssel kennt. Zum Beispiel, weil $Z$ der Hersteller seines Betriebssystems ist und im mitgelieferten Web-Browser eingestellt ist, dass $Z$ vertraut werden kann und ebenfalls $Z_{pub}$ hinterlegt wurde.

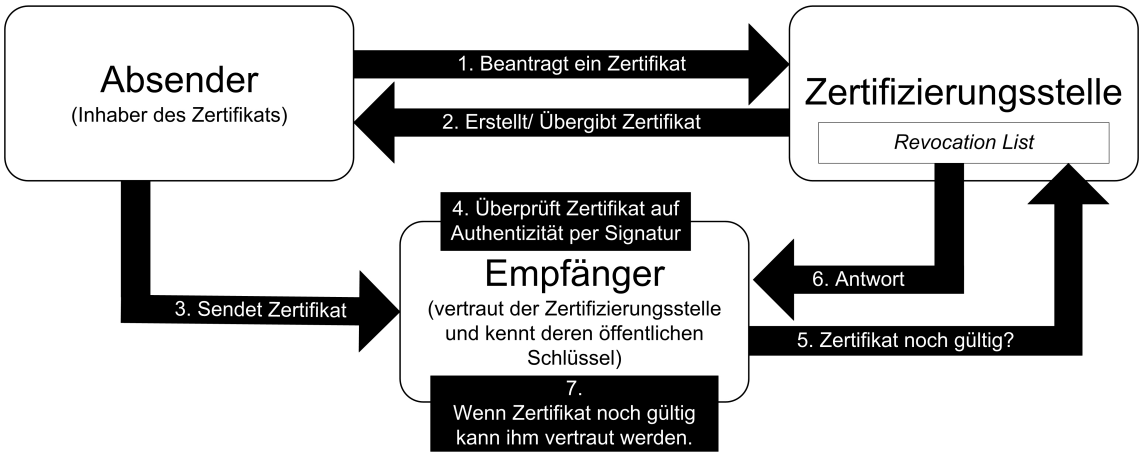


Abbildung 1.5: Einfaches Szenario für Zertifikatsvergabe und -verteilung

Nun möchte $A$ sicher Nachrichten an $E$ verschicken und muss ihm zu diesem Zweck glaubwürdig $A_{pub}$ übermitteln. Dazu läßt sich $A$ von $Z$ ein Zertifikat mit oben angegebenem Inhalt erstellen und persönlich übergeben. Dieses Zertifikat kann er nun an $E$ senden und ihm somit u.a. seinen öffentlichen Schlüssel mitteilen.
Über die von $Z$ ausgestellte Signatur kann $E$ nun die Echtheit des Zertifikats feststellen. Abschließend muss $E$ prüfen,ob das Zertifikat noch gültig ist. Dies geschieht zum einen über die im Zertifikat angegebene Gültigkeitsdauer und über die *Revocation List* der Zertifizierungsstelle, in der sämtliche von ihr ausgestellten und vor Ablauf der Gültigkeitsdauer unglaubwürdig gewordenen Zertifikate aufgeführt sind.
Falls all diese Maßnahmen zum Erfolg führen kann $E$ allen Nachrichten von $A$ vertrauen (regelmäßige Prüfung der Revocation List vorausgesetzt).

Dieser Spezialfall tritt natürlich in den seltensten Fällen auf, da im Prinzip jeder eine eigene Zertifizierungsstelle einrichten und Zertifikate vergeben kann.
Diese Zertifizierungsstellen können wiederum von anderen Stellen zertifiziert werden, was große Zertifizierungshierarchien entstehen läßt.
Wenn ein Benutzer nun ein von einer solchen, ihm unbekannten, Stelle ausgestelltes Zertifikat erhält kann er ihm nur so stark vertrauen, wie er der Stelle selbst vertraut. Dazu

muss er zuerst den öffentlichen Schlüssel zur Überprüfung der Signatur erhalten, möglicherweise über ein anderes Zertifikat, welches ebenfalls evaluiert werden muss.
Auf diesem Wege wird die gesamte Hierarchie durchlaufen, bis entweder eine Stelle erreicht wird, der bereits das Vertrauen ausgesprochen wurde (Abb.1.6) oder deren Spitze erreicht ist.
Im zweiten Fall steht der Empfänger vor der Entscheidung, dem öffentlichen Schlüssel dieser Stelle zu vertrauen, dazu muss dieser auf anderem Wege sicher übertragen werden, oder diesen abzulehnen, was zur Folge hat, dass auch das Zertifikat des ursprünglichen Absenders nicht als vertrauenswürdig angesehen wird.

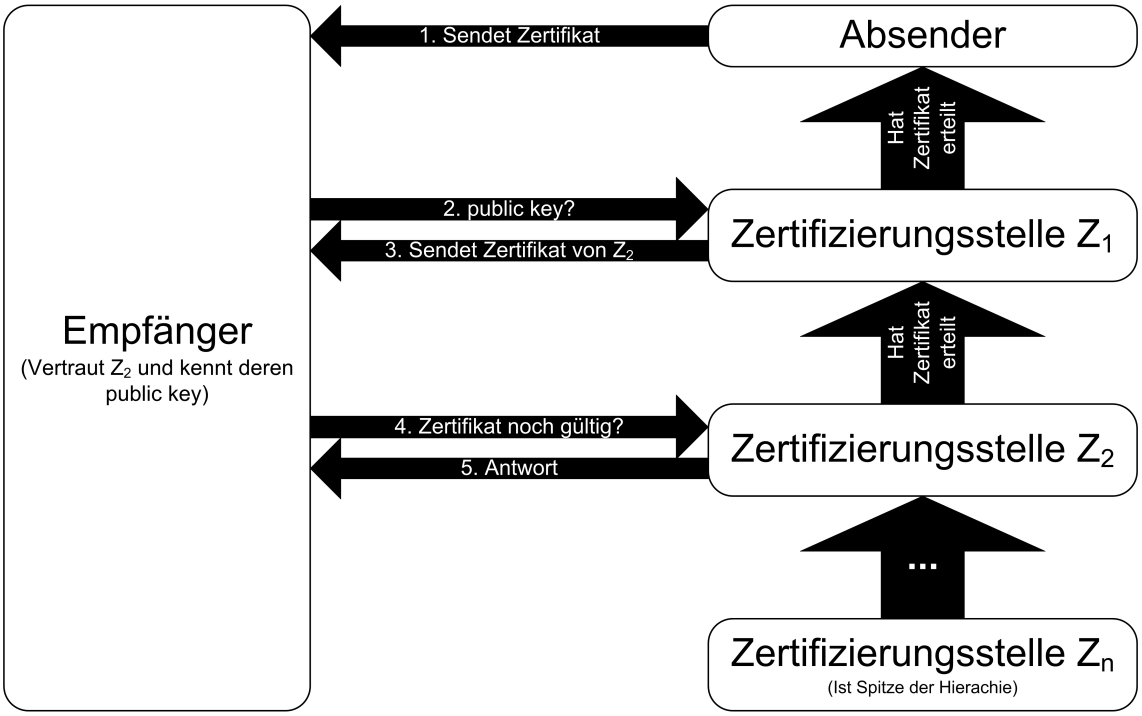


Abbildung 1.6: Zertifikatsvergabe und -verteilung in Zertifizierungshierarchie

Hier ist leicht zu erkennen, dass diese *PKI* (Public Key Infrastructure) einen hohen organisatorischen und materiellen Aufwand erfordert, was zur Folge hat, dass Zertifikate insbesondere zur kommerziellen Anwendung sehr teuer sein können.

Sicherheit ist also nicht umsonst zu haben und es muss stets zwischen den Sicherheitserfordernissen und materiellen Möglichkeiten abgewogen werden.

## 1.3 Zusammenfassung

Grundsätzlich kann angesichts der technischen Möglichkeiten davon ausgegangen werden, dass theoretisch ein sehr großer Grad an Sicherheit möglich ist. Meist ist die Entwicklung von Sicherheitsmechanismen schneller, als die der Angriffsmöglichkeiten. Zum Beispiel im Bereich der Kommunikationssicherheit durch asymmetrische Verschlüsselungsverfahren, die nur sehr schwer zu „knacken“ sind.

Doch letztendlich gibt der Benutzer selbst den Ausschlag, wie sicher das System ist. Gegen Social Engineering oder Leichtsinn können oft nur organisatorische Maßnahmen ergriffen werden. Auch der Schutz privater Schlüssel vor unbefugtem Zugriff Fremder muss unbedingt gewährleistet sein.
Gelangt ein Angreifer ins System, weil seine Viren oder er selbst sich Zutritt verschafft haben oder er ihnen fahrlässig bzw. vorsätzlich gewährt wurde kann der Schaden enorme Ausmaße annehmen.

Sehr treffend formuliert wurde dies in einem Zitat von *R.H. Baker*[1]:

*„The real challenges are human, not technical. Oldtimers will recognize a once popular saying that the most important part of an automobile is the nut that holds the steering wheel. That´s still true, even though a modern steering wheel may also contain an air bag and any number of controls and theft devices.“*

# Literaturverzeichnis


[1] **Opplinger, Rolf:** IT-Sicherheit - Grundlagen und Umsetzung in der Praxis, vieweg 1997

[2] **Tanenbaum, Andrew S.:** Moderne Betriebssysteme 2. überarbeitete Auflage, Pearson Studium 2003

[3] **Minas, Mark:** Skriptum zur Vorlesung *Sichere Systeme* HT 2004, Universität der Bundeswehr München

[4] www.wikipedia.de

# Kapitel 2

# Authentifizierung in Windows Betriebssystemen

*Carsten Schulz*

*In diesem Abschnitt dreht sich alles um die sicherheitsrelevanten Bereiche Authentifizierung und Autorisierung in den Betriebssystemen von Microsoft. In einem kurzen Überblick wird zunächst die Versionsgeschichte der Windows Familie aufgezeigt. Anschließend werden erste Verfahren zur Authentifizierung besprochen, welche ihre Anwendung in Windows NT 4.0 fanden. Anhand von Windows 2000 und des darin integrierten Verzeichnisdienstes Active Directory werden dann die aktuell implementierten Authentfizierungsmechanismen ausführlich dargestellt. Die Versionen Windows XP und Server 2003 werden in dieser Arbeit nicht weiter betrachtet, da sie bezüglich Authentifizierung keine Neuerungen gegenüber Windows 2000 aufweisen. In einem kurzen Ausblick auf den XP-Nachfolger Longhorn werden abschliessend zukünftige Sicherheits- und Authentifizierungsmechanismen angesprochen.*

## Inhaltsverzeichnis

## 2.1 Einleitung und Motivation

Redet man von IT-Sicherheit im Zusammenhang mit Windows, so kann es öfter vorkommen, dass sich ein Schmunzeln im Gesicht des Gegenüber breit macht. Regelmäßig kann man in den einschlägigen Fachzeitschriften Berichte über Sicherheitslücken in Windows lesen. Windows ist zwar das meistgenutzte Betriebssystem, doch hat es in der Tat hat den Ruf, unsicher zu sein. Weiterhin scheint es so, als würde die Unix-Welt jede Sicherheitslücke von Windows wie einen Etappensieg feiern. Fakt ist, dass auch Unix nicht perfekt ist. Eine heiß-diskutierte Sicherheits-Studie kam beispielsweise zu dem Ergebnis, dass Microsoft Server 2003 im Vergleich mit einem Red Hat Enterprise Linux-Server eindeutig vorne liegt. In dieser Arbeit soll die Windows-Sicherheit bzw. die Mechanismen, die für eine etwaige Sicherheit in Windows sorgen, unvoreingenommen betrachtet und aufgezeigt werden.

## 2.2 Microsoft Windows – von 1985 bis heute

Nachdem Microsoft 1981 mit MS-DOS 1.0 begonnen hatte, Betriebssysteme für Computer zu entwickeln, kam 1985 mit Windows 1.0 eine zweite Produktlinie hinzu. Diese war zunächst für den Einzelplatz (Home Edition) konzipiert und bekam erst später eine Netzwerkunterstützung. Aus der dritten Produktlinie, welche 1987 mit dem MS-OS/2 1.0 begann und für den professionellen Einsatz gedacht war, entstand schließlich im Jahr 1993 Windows NT 3.1 (NT = New Technology). In der hierfür benötigten Entwicklungszeit von 5 Jahren entstanden ca. 3,1 Millionen Codezeilen. Die im August 1996 erschienene Version NT 4.0 beinhaltete bereits 16 Millionen Codezeilen. Windows NT war die erste Microsoft-Plattform, die als halbwegs sicher anzusehen war (siehe nächstes Kapitel). Für die Entwicklung des Nachfolgers Windows 2000 benötigte man schätzungsweise 3 Jahre und etwa 1 Milliarde Dollar. In 30 Millionen Codezeilen wurden ungefähr 10.000 Bugs aus vorangegangenen Versionen behoben. Seit Oktober 2001 ist Windows XP auf dem Markt, welches heute das – vor allem im privaten Bereich – weltweit meistgenutzte Betriebssystem darstellt. XP sollte die vorherige Spaltung von Consumer- und Business- Windows wieder vereinen. Schließlich wurde 2003 die Version Windows 2003 Server als Weiterentwicklung der 2000er Datacenter Server-Version auf den Markt gebracht. Für Mitte 2006 ist der XP-Nachfolger Longhorn geplant, welcher einige neue Sicherheitsfeatures beinhalten wird. Folgende Tabelle zeigt die wichtigsten Versionen von Microsoft Windows im Überblick.

| **Jahr** | **Windows-Version** |
|---|---|
| 1985 | 1.0 für 8088 CPUs |
| 1993 | for Workgroups 3.11 (rudimentär netzwerkfähig) |
| 1993 | NT 3.1 und Advanced Server |
| 1996 | NT 4.0 und NT Terminal Server Edition |
| 1995 | 95 Version A |
| 1997 | NT Server 4.0 Enterprise Edition |
| 1999 | 98 SE |
| 2000 | 2000 und 2000 Datacenter Server |
| 2001 | XP (whistler) |
| 2003 | Server 2003 (whistler server) |
| 2006 | Longhorn |

Tabelle 2.1: einige Versionen von Windows im Überblick

## 2.3 MS Windows NT 4.0

Das fast vollständig in der Programmiersprache C geschriebene 32-Bit-Betriebssystem Windows NT stellte zur Zeit der Betriebseinführung ein – zumindest für Microsoft – vollkommen neues Konzept dar. Es ist für Einzelplatzrechner und als Trägerbetriebssystem für ein Netzwerk gleichermaßen einsetzbar. Man unterscheidet zwischen der Workstation-Edition, welche für kleine Netzwerkumgebungen (maximal 10 Client-Verbindungen) einige wenige Netzwerkfeatures enthält und der Server-Variante, die mit zahlreichen Netzwerkfunktionen aufwartet. Durch einen preemptiven Multitasking-Betrieb bietet Windows NT eine Plattform für Client-Server Anwendungen. Durch einen Remote Access Service ermöglicht es die Einwahl von bis zu 256 Clients.

Des Weiteren unterstützt NT das alte DOS-Dateisystem FAT (File Allocation Table), das von OS/2-Dateisystem HPFS (High Performance File System), sowie NTFS (New Technology File System), welches im Vergleich zu den anderen beiden eine viel höhere Leistungsfähigkeit, Robustheit und weitaus höherer Sicherheitseigenschaften bietet. NTFS ermöglicht es beispielsweise, individuelle Zugriffsrechte für Dateien und Verzeichnisse an Benutzer und Gruppen zu vergeben.

### 2.3.1 Sicherheitskomponenten

Windows NT verfügt über folgende Sicherheitskomponenten:

- Local Security Authority (LSA)
- Security Account Manager (SAM)
- Security Reference Monitor (SRM)

Die Local Security Authority dient der Benutzeridentifikation und -authentifikation. Sie überprüft also, ob ein Anwender auf das Betriebssystem zugreifen darf. Zudem verwaltet die LSA die lokalen Sicherheitsrichtlinien sowie Überwachungsrichtlinien. Mit Hilfe der vom SRM gelieferten Meldungen ist sie dafür zuständig, die Protokolldaten zu erzeugen.

Der Security Account Manager pflegt jene Datenbank, in welcher die Informationen (z.B. Zugriffsrechte) über alle Benutzer- und Gruppenkonten gesammelt werden, die sogenannte SAM-Datenbank. Meldet sich ein Benutzer im Systen an, so sorgt der SAM dafür, dass die eingegebenen Daten (Benutzername und Passwort) mit den Einträgen in der Datenbank verglichen werden.

Der Security Reference Monitor überwacht schließlich die Einhaltung der von der LSA vergebenen Zugriffsberechtigungen (z.B. auf Dateien und Ordner), sowie die Ausführung der angeforderten Aktionen. Er kann weiterhin, wie oben bereits erwähnt, die Meldungen für die Protokolldateien erzeugen.

### 2.3.2 Authentifizierung mit dem NT-LAN-Manager

Seit jeher findet die Authentifizierung in einem Windows-System über die Abfrage von Benutzerkennung und Passwort statt. In Windows NT erfolgt diese über den sogenannten NT-LAN-Manager, dem NTLM. Während man bei dessen Vorgänger, dem LAN-Manager unter Windows 3.11 kaum von einer geschützten Übergabe von Benutzernamen und Benutzerpasswort sprechen kann, verfügt die NTLM-Variante bereits über ein – zur Zeit der Betriebseinführung – erstaunliches Maß an Sicherheit. Dies erfolgt durch eine verschlüsselte Übergabe, wobei die Codierung nach heutigem Stand als ein wenig unsicher anzusehen ist.

Die NTLM-Authentifikation basiert zunächst einmal auf einem Kennwort, das in der Security Account Datenbank des Systems oder des Domänencontrollers gespeichert ist. Aus Kompatibilitätsgründen wird für jeden Anwender das Passwort zweimal abgelegt. Einerseits für die Unterstützung alter LAN-Manager-Authentifikationen und zum anderen für die NT 4.0-Variante. Erstere ist hierbei der unsichere Faktor der Passwortspeicherung. Hier wird mit dem DES-Algorithmus aus einer auf 14 Zeichen begrenzten Klartextlänge ein Hash errechnet. Ein Hash ist eine Zeichenkette fixierter Länge, die durch das Verschlüsseln mithilfe einer mathematischen Funktion, unabhängig von der Länge des Ausgangstextes, entsteht und nicht wieder entschlüsselt werden kann (unumkehrbar). Das Betriebssystem selbst verwendet bei der Eingabe eines Kennwortes genau dieselbe Funktion und vergleicht dann zur Authentifikation den eben errechneten Hash-Wert mit dem in der SAM-Datenbank gespeicherten Wert. Stimmen beide Werte überein, war die Authentifizierung erfolgreich. Um potentielle Angreifer zu verwirren, wird das bereits verschlüsselte Kennwort nochmal über eine spezielle Anwenderkennung (RID) verschlüsselt. Dadurch wird erreicht, dass man das Kennwort nicht mehr direkt dem Anwender zuordnen kann. Im Wesentlichen benötigt man lediglich entsprechende, administrative Rechte auf das Kennwort und darüber hinaus Zugriff auf den Verschlüsselungsalgorithmus selbst, um die letzte Verschlüsselung nachzuvollziehen. Daher ist diese eher als eine zusätzliche Sicherheitsvorkehrung anzusehen als eine echte Verschlüsselung.

Bei der zweiten, der NT-Variante handelt es sich um ein deutlich besser verschlüsseltes Kennwort. Obwohl theoritsch bis zu 128 Zeichen möglich sind, wird aus design-technischen Gründen auch hier das Klartextkennwort auf 14 Zeichen begrenzt. Dieses wird dann mithilfe des RSA-MD4-Algorithmus, welcher eine deutlich höhere Verschlüsselungstiefe als der DES erreicht, verschlüsselt. Auch hier handelt es sich um einen errechneten Hash, welcher bei der Authentifikation vom Betriebssystem verglichen wird.

Die NTLM-Authentifizierung basiert auf einer Unterteilung der lokalen Sicherheit (LSA) in zwei Teile. Gemäß der Funktionsweise eines Challenge-Response-Verfahrens findet der erste Teil der Authentifizierung auf dem Computer statt, an dem sich der Benutzer anmeldet. Der zweite Teil findet hingegen auf dem Computer statt, auf welchem das Benutzerkonto abgelegt ist. Handelt es sich um eine NT-Workstation-Version, ist es also ein und derselbe Computer, so ist diese Unterteilung bedeutungslos. Befindet sich der Computer, auf dem sich der Anwender anmeldet, allerdings in einer Domäne mit entsprechendem Domänencontroller (DC), so bekommt die Unterteilung eine andere Bedeutung. Hier findet der zweite Teil der Authentifizierung auf dem Domänencontroller statt. Wie in Abbildung 2.1 ersichtlich übergibt der Client-Rechner hierzu die Authentifizierungsdaten an den Anmeldedienst, der die Weiterverarbeitung am DC gewährleistet. In einer komplexen Netzwerkumgebung kann diese Authentifizierungsmethode durchaus zu einer hohen Netzwerkbelastung führen. Dies rührt daher, dass beispielsweise bei eine Anmeldung von einem Client-Rechner an einem Dateiserver, dieser erst eine Anfrage an den Domänencontroller starten muss, um festzustellen, ob der entsprechende Benutzer für den Zugriff auf den Dateiserver autorisiert ist. Kommt es vor, dass eine Authentifizierung (z.B. durch Leerlauf) abläuft, so müssen die Verhandlungen erneut angestoßen werden.

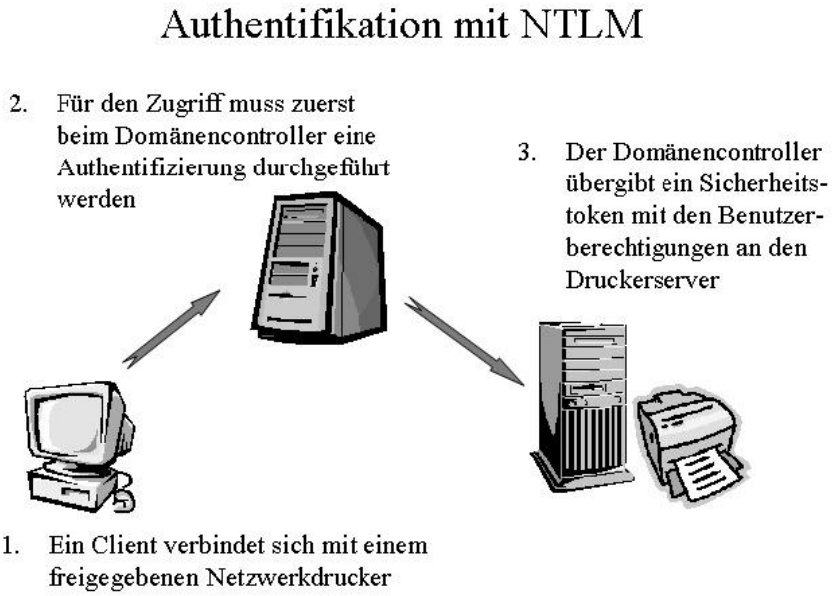


Abbildung 2.1: NTLM-Authentifizierung [Quelle: „Windows Sicherheit“, Addison Wesley Verlag]

## 2.4 Windows 2000

Die Windows 2000-Betriebssystemfamilie besteht aus den Versionen „Professional“, „Server“, „Advanced Server“ und „Datacenter Server“. Sie unterscheiden sich hauptsächlich in den Leistungsmerkmalen für die Unterstützung von Multiprocessing, Festplattenverwaltung (Clustering), sowie Lastenausgleich beim Einsatz mehrerer Windows 2000 Server (siehe folgende Tabelle).

| **Version** | **Funktion** |
|---|---|
| Professional | für Arbeitsstationen und Netzwerkclients,<br>ersetzt Windows NT 4.0 Workstation |
| Server | entwickelt, um anderen Systemen im Netzwerk Dienste und Ressourcen bereitzustellen,<br>ersetzt NT 4.0 Server,<br>unterstützt maximal zwei CPUs |
| Advanced Server | erweitert um Funktionen zur Unterstützung des Lastausgleichs und Clustering,<br>unterstützt Speicherkonfigurationen bis zu 64 GB,<br>kann bis zu vier CPUs verwalten |
| Datacenter Server | der umfangreichste Windows Server,<br>unterstützt fortschrittlicheres Clustering als Advanced Server und<br>kann bis zu 16 CPUs verwalten |

Tabelle 2.2: Windows 2000 Versionen im Überblick

Windows 2000 ist sehr viel leistungsfähiger und sicherer als NT 4.0, was durch eine Vielzahl neuer Funktionen, vor allem neuer Systemmanagement- und Verwaltungskonzepte, erreicht wird. Neben schon altbekannten Sicherheitsvorkehrungen gehörten nunmehr Features wie IP Security (IPSec), CryptoAPI, SSL 3.1, Kerberos-Beglaubigung, SmartCard, Public Key Infrastructure, Encrypting File System (EFS), sowie Analyse und automatische Konfiguration sicherheitsrelevanter Einstellungen zum Reservoir des neuen Microsoft-Betriebssystems. Das von Windows NT bekannte Domänenmodell wurde weiterentwickelt, trägt nun den Namen Active Directory und bildet den Kern der Systemsicherheit von Windows 2000.

### 2.4.1 Active Directory

Active Directory ist der Verzeichnisdienst für Windows 2000 Server. Er kann als eine Datenbank verstanden werden, in der Informationen zu Objekten gespeichert werden. Diese Objekte repräsentieren Benutzer, Gruppen, Anwendungen, Dateien sowie Drucker, Computer und weitere Peripheriegeräte. Gleichzeitig stellt er diese Informationen Benutzern und Administratoren über einfache Suchfunktionen zur Verfügung. Active Directory bietet Sicherheitsfunktionen wie die Anmeldeauthentifizierung und den gesteuerten Zugriff auf Verzeichnisobjekte. So wird es Administratoren ermöglicht, mit einer einzigen Netzwerkanmeldung Verzeichnisdaten und Verzeichnisstruktur im gesamten Netzwerk zu

verwalten. Netzwerkbenutzer können dadurch ebenfalls auf benötigte Ressourcen netzwerkweit zugreifen. Die richtlinienbasierte Verwaltung erleichtert es, selbst komplexere Netzwerke zu überwachen.

Um Active Directory zu installieren, benötigt man mindestens einen Computer mit der Version Windows 2000 Server. Dieser kann dann mit dem Programm „dcpromo“ zu einem Domänencontroller heraufgestuft werden. Sämtliche Konto- und Richtlinieninformationen des Verzeichnisdienstes werden auf dem Domänencontroller gespeichert.

### Komponenten von Active Directory

Active Directory zeichnet sich durch folgende Komponenten aus:

- Datenbank:
  Das AD besitzt eine Datenbank, die alle Informationen über das Netzwerk wie Benutzer, Gruppen und Computer enthält. Die Datensätze nennt man Objekte, deren Eigenschaften Attribute. Das wichtigste Attribut eines Objektes ist hierbei immer der Global Unique Identifier (GUID). GUIDs sind 128 Bit breite, weltweit eindeutige Werte, die ein Objekt identifizieren.

- Schema:
  Die Gesamtheit aller Regeln, die für die erfolgreiche Speicherung von Daten eingehalten werden müssen, nennt man Schema. Definiert wird die Struktur und der Inhalt des ADs, einschließlich aller Attribute, Objekte und Objektklassen. Ein Standardschema wird bei der Neu-Installation von Active Directory (bzw. des ersten Domänencontrollers) erzeugt und kann dann abgeändert oder beliebig ergänzt werden. Der Zugriff auf das Gesamtschema von AD ist über die DACL (Discretionaly Access Control List) gesteuert, so dass nur autorisierte Accounts Schemaänderungen vornehmen können.

- Replikationsdienst:
  Der Replikationsdienst ist einer der wichtigsten Mechanismen in Active Directory. Um Redundanz der Daten zu gewährleisten, repliziert (spiegelt) dieser Dienst das Verzeichnis auf andere Server. Hierdurch erreicht man zudem eine höhere Ausfallsicherheit des Verzeichnisdienstes.

- Abfragemechanismus:
  Die Suche nach Ressourcen ist eine der Hauptaufgaben eines Verzeichnisdienstes. Diese wird bei Active Directory durch den Global Catalog (GC) und das LDAP-Protokoll (Lightweight Directory Access Protocol) realisiert. Der ebenfalls auf dem Domänencontroller gespeicherte Globale Katalog enthält eine Kopie der wichtigsten Objekt-Attribute des Verzeichnisses. Er speichert zwar standardmäßig nicht alle Objekte des Verzeichnisses, jedoch immer alle Objekte mit allen Attributen seiner Domäne. Zusätzlich werden noch die Attribute von allen Objekten der Gesamtstruktur, bei denen in der Schemadefinition das entsprechende Flag gesetzt ist, gespeichert. Der Abfragemechanismus benutzt LDAP, um Objekte zu addressieren. LDAP setzt direkt auf den TCP/IP-Protokollstack auf. Der Vorteil hiervon ist, dass

ein große Menge an Daten wegfällt, welche aus der Darstellungs- und der Sitzungschicht des OSI-Modells kommen würden. Daher ist LDAP sehr leistungsfähig und stellt nicht zuletzt deshalb mittlerweile des Standardprotokoll für den Zugriff auf Verzeichnisdienste dar.

- Sicherheitskonzept:
  Das Verzeichnis ist Teil der Windows 2000 Trusted Computing Base und nimmt uneingeschränkt an der Sicherheitsinfrastruktur von Windows 2000 teil. Kerberos, ein Microsoft-unabhängiger Mechanismus, wird für die Authentifizierung über das Netzwerk benutzt. Alle Objekte und Attribute in Active Directory werden durch ACLs (Access Control Lists) geschützt. ACLs sind Listen, in der an Benutzer verbindlich vergebene Rechte gespeichert werden. Die Zugriffsüberprüfungsroutinen von Windows 2000 verwenden die ACLs, um die Gültigkeit jedes versuchten Zugriffs auf ein Objekt oder Attribut in Active Directory zu überprüfen. Ein Vorteil von AD ist die Möglichkeit, administrative Rechte „aufzuteilen“. Neben Benutzern und Administratoren gibt es weiterhin einige Ebenen dazwischen, die es ebenfalls ausgewählten Benutzern erlauben, beispielsweise Passwörter anderer Benutzer zurückzusetzen.

### Aufbau und Struktur des Active Directory

Beim Active Directory muss man zwischen der physischen Struktur und der logischen Struktur unterscheiden. Die logische Struktur baut sich wie folgt auf:

- Die Domäne:
  Sie ist die Kerneinheit von Active Directory. Die Domäne ist eine logische Verwaltungseinheit, die der Strukturierung des Netzwerkes dient. Organisationen partitionieren so die Gesamtstruktur (siehe weiter unten), um zu verhindern, dass Daten auf Standorte repliziert werden, an denen sie nicht erforderlich sind. Auf diese Weise kann ein Netzwerk mit eingeschränkter Bandbreite global skaliert werden. Die Domäne stellt in Active Directory die Verwaltungsgrenze für Objekte, Benutzer, Gruppen und Computer dar. Man unterscheidet zwischen einheitlichen (nativen) Domänen und Mixed-Mode-Domänen. Während erstere ausschließlich aus Windows 2000-Domänencontrollern bestehen, beinhalten Mixed-Mode-Domänen auch die älteren NT 4.0-Domänencontroller. Da die Abwärtskompatibilität gewährt bleiben muss, muss man bei Mixed-Mode auf die Vorteile von Windows 2000 verzichten, so zum Beispiel auf die Verwendung der neuen Gruppentypen. Wird bei einer Neuinstallation ein neues AD angelegt, so ist dies zunächst einmal im Mixed-Mode. Wird nicht geplant, NT 4.0- Domänencontroller ins Verzeichnis aufzunehmen, bzw. sind zu einem späteren Zeitpunkt alle NT-Domänencontroller ersetzt, so kann einfach in den nativen Modus umgeschaltet werden. Eine Rückkehr in den Mixed-Mode ist danach allerdings nicht mehr möglich. Eine Domäne braucht immer mindestens einen Domänencontroller. Mehrere DCs sind möglich und im Sinne von Redundanz und Ausfallsicherheit sogar anzuraten.

- Struktur und Gesamtstruktur:
  Gesamtstrukturen, auch Forests oder Wälder genannt, bestehen aus Strukturen,

auch Trees oder Bäume genannt (Abbildung 2.2). Strukturen wiederum bestehen aus Domänen (Abbildung 2.3). Dies erlaubt den Aufbau einer beliebig großen baumartigen Hierarchie von Domänen , in dessen Verbund Ressourcenzugriffe und Suchaktionen möglich sind. Domänen einer Struktur werden automatisch mit transitiven beidseitigen (Kerberos-) Vertrauensstellungen miteinander verbunden. Dadurch ist ein strukturweiter, ja sogar ein gesamtstrukturweiter Ressourcenzugriff möglich. An der Spitze dieser Hierarchie steht die Struktur-Stammdomäne (Masterdomäne), deren Namen nicht veränderbar ist. Der erste Domänencontroller einer Struktur wird automatisch Strukturstamm. Auch die Gesamtstruktur besitzt einen Stamm, welcher ebenfalls dem ersten Domänencontroller in der Struktur zugewiesen wird. Um eine gesamtstrukturweite Replikation sowie den gesamtstrukturweiten Ressourcenzugriff zu ermöglichen, unterhalten alle Strukturstämme beidseitig transitive Vertrauenstellungen zum Gesamtstruckturstamm. Innerhalb einer Gesamtstruktur wird ein einziges Schema und ein einziger Globaler Katalog verwendet.

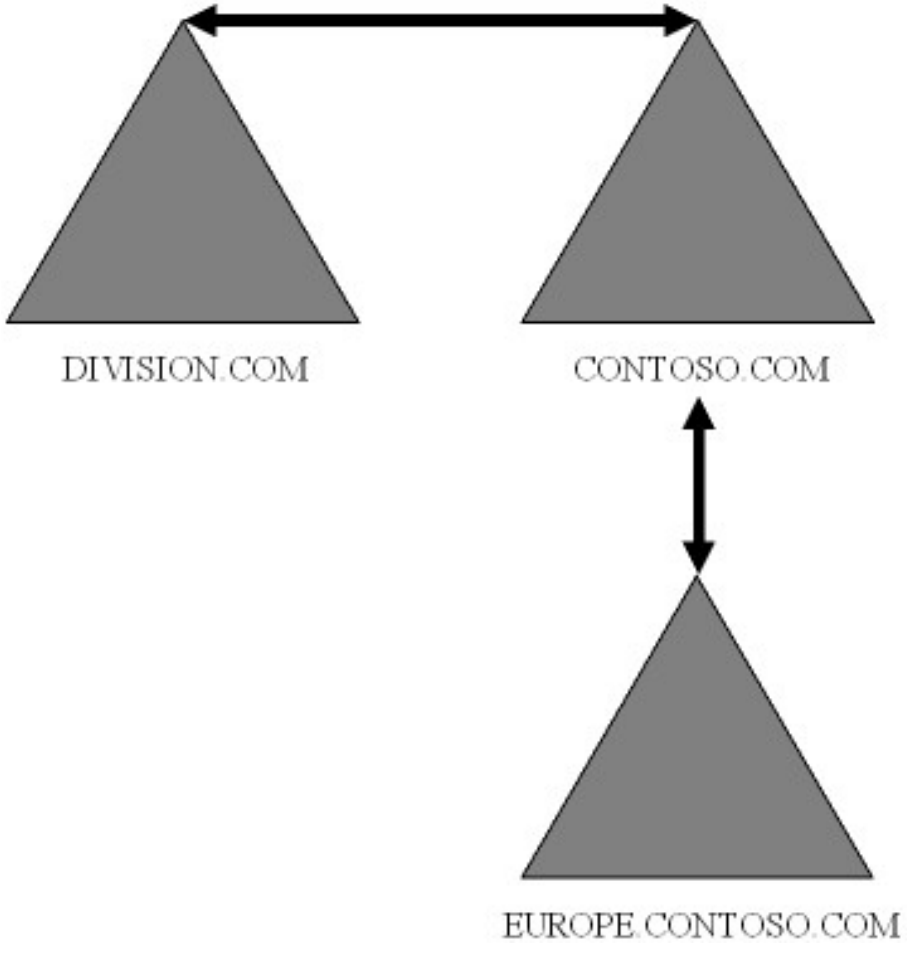


Abbildung 2.2: Beispiel für eine AD-Gesamtstruktur

- Vertrauensstellungen:
  Domänen werden mithilfe von sogenannten Vertrauensstellungen verbunden, um Zugriffe von Konten einer Domäne auf Ressourcen einer anderen Domäne zu gewährleisten. Notwendig wird dies, da Domänen eine Sicherheitsbarriere darstellen.

Alle Vertrauensstellungen bei AD sind im Gegensatz zu früheren Windows NT-Vertrauenstellungen per Voreinstellung beidseitig und transitiv. Das bedeutet unter anderem, dass das Vertrauen „durchgereicht“ werden kann. Vertrauensstellungen werden automatisch zwischen allen Domänen einer Gesamtstruktur eingerichtet und können nicht unterbunden werden. Jeder Mitgliedcomputer kann alle Benutzer oder Gruppen anderer Domänen in der Gesamtstruktur erkennen und diesen Zugriff gewähren. Vertrauensstellungen mit Domänen in anderen Gesamtstrukturen können individuell eingerichtet werden.

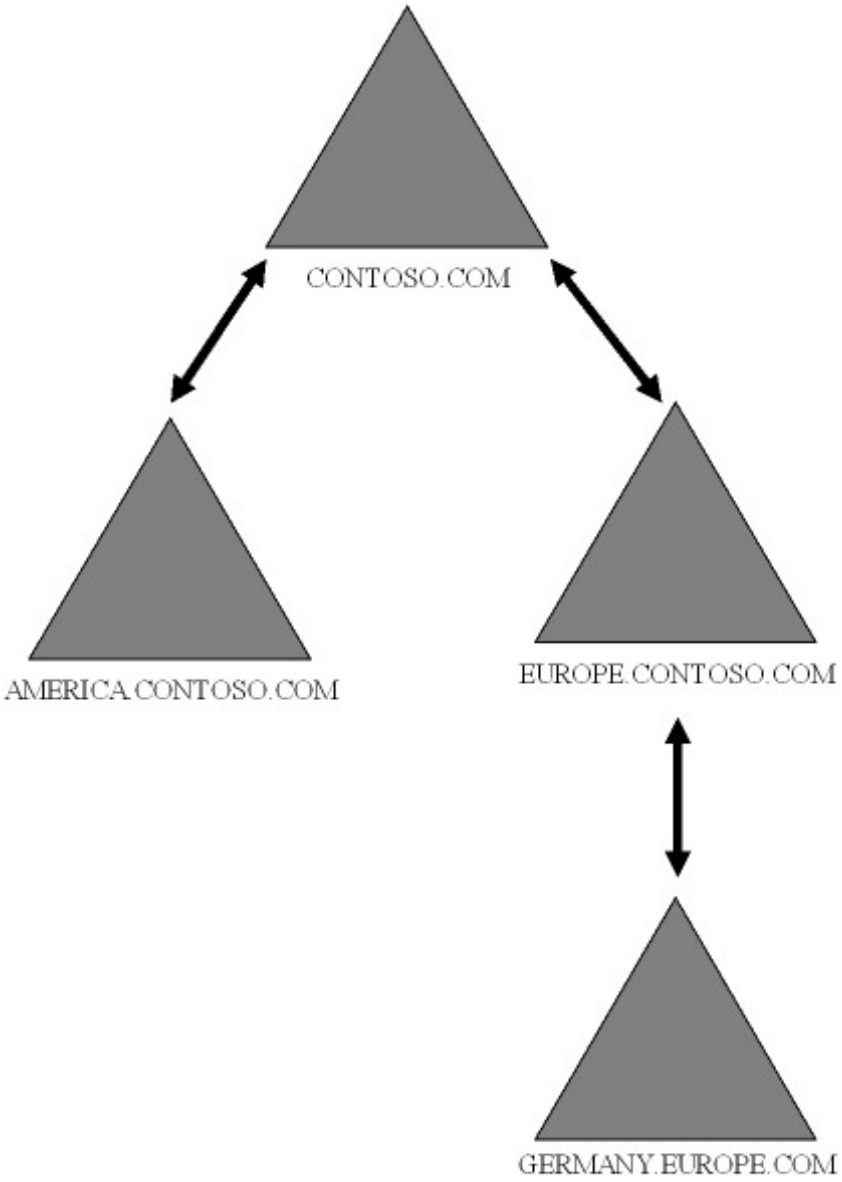


Abbildung 2.3: Beispiel für eine AD-Struktur: Die Domänen verfügen über gegenseitige Vertrauensstellungen

- Organisationseinheiten:
  Organisationseinheiten (OUs = Organizational Units) stellen ein weiteres Element zur logischen Gliederung im AD dar. Eine OU ist ein Containerobjekt, welches Res-

sourcen innerhalb einer Domäne zusammenfasst. So können sie einfach von einem, nur für diese OU zuständigen, Administrator verwaltet werden. Eine OU könnte beispielsweise Benutzer einer bestimmten Abteilung mit ihren Computern und den Freigaben, die sie verwenden, enthalten. Eine andere könnte Drucker und die dazugehörigen Freigaben zusammenfassen. Es wird also eine sehr feine Aufteilung zur Ressourcenverwaltung ermöglicht. Innerhalb einer OU können zudem bestimmte Rechte an beliebige Benutzer vergeben werden.

- Container:
  Alle Objekte (nicht nur OUs) im AD sind Container. Ein Container repräsentiert kein physisches Objekt wie ein Benutzer oder ein System. Stattdessen verwendet man ihn nur zur Gruppierung anderer Objekte. Container-Objekte können innerhalb anderer Container verschachtelt werden. Auf die selbe Weise, wie ein Dateiordner Dateien und Dokumente aufnimmt, kann ein Verzeichnis-Container Verzeichnis-Objekte aufnehmen. Somit ist es bspw. möglich, einen Drucker als Kind eines Computers anzulegen.

Die physikalische Struktur von Active Directory wird im Wesentlichen durch Standorte (Sites) definiert. Ein Standort ist ein schnelles, zuverlässiges Netzwerk, für das keine nutzungsabhängigen Kosten entstehen. Ein LAN oder eine Gruppe von LANs, die durch einen Hochgeschwindigkeits-Backbone verbunden sind, können als Standort angesehen werden. Man spricht auch von einer Zusammenfassung von IP-Teilnetzen oder Subnetzen. Standorte werden hauptsächlich eingerichtet, um die Netzwerkbandbreite zu optimieren. Beispielsweise kann zwischen Standorten die Replikationstätigkeit gesteuert werden, welche innerhalb eines Standortes jederzeit spontan bei einer Änderung stattfindet. Standorte können zahlreiche Domänen enthalten, da es sich um einen physikalischen bzw. Netzwerkstandort und nicht um einen logischen Standort handelt. Umgekehrt kann sich eine Domäne über mehrere Standorte erstrecken. Die zentrale Frage bei der Planung ist die Netzwerkgeschwindigkeit zwischen den Mitgliedern einer Gesamtstruktur und die damit verbundenen Kosten. Verbunden werden die Standorte durch sogenannte Standortverknüpfungen (Sitelinks). Sie stellen - oft unzuverlässige - Verbindungen zwischen den Sites dar, welche meistens niedrigere Übertragungsgeschwindigkeiten aufweisen (Kostengründe!). Ein WAN (Wide Area Network), das zwei schnelle Netzwerke verbindet, ist ein Beispiel für eine Standortverknüpfung.

### Authentifizierung und Autorisierung

Windows 2000 setzt die Überprüfung der Identität eines jeden Benutzers voraus, bevor ihm Zugriff auf Netzwerkressourcen erlaubt wird. Diese Authentifizierung ist Teil des Anmeldevorgangs. Sie umfasst die Identifikation gegenüber dem Sicherheitssystem anhand einer eindeutigen Benutzer-ID. Die Gewährung oder Ablehnung des Zugriffs auf die Ressource wird als Autorisierung bezeichnet.

Identitätsdaten und Anmeldedaten (z.B. Kennwörter) der Benutzer werden vom AD in Form eines Benutzerkontenobjekts gespeichert. Für jedes Kontenobjekt, welches authentifiziert werden kann, wird eine eindeutige Sicherheits-ID (SID) erstellt. Sie enthält unter

anderem die ID der Domäne, in der sich das Objekt befindet. Objekte mit einer SID (z.B. Benutzer, Gruppen und Computer) werden auch Sicherheitsprinzipale (Security-Principals) genannt. Eine Benutzeridentifizierung kann nur von dem Domänencontroller durchgeführt werden, auf dem sich das Konto des Benutzers befindet. Wird ein Benutzer erfolgreich autorisiert, dann wird vom Sicherheitsystem auf dem authentifizierenden Domänencontroller ein Teil der notwendigen Autorisierungsdaten generiert. Diese Daten setzen sich aus der primären (stets eindeutigen) Sicherheits-ID (SID) des Benutzers sowie den SIDs der Gruppen, zu denen der Benutzer gehört, zusammen. Die restlichen Autorisierungsdaten werden dann generiert, wenn auf eine Netzwerkressource zugegriffen wird. Dabei wird ein Zugriffstoken generiert, bestehend aus den eben angesprochenen SIDs und den Berechtigungen des Nutzers auf dem lokalen Computer. Zugriffstoken werden von allen Ressourcen im Netzwerk anerkannt.

Die Autorisierung erfolgt unter Zuhilfenahme einer „Discretionary Access Control List“ (DACL), welche jedem gesicherten Objekt im Active Directory zugewiesen wurde. Sie beinhaltet die Zugriffsrechte von Benutzern und Gruppen auf das jeweilige Objekt. Bei der Zugriffsüberprüfung des Sicherheitssystems wird das Zugriffstoken gegenüber der entsprechenden DACL ausgewertet und ermittelt, auf welcher Ebene der Benutzer Zugriff auf die Netzwerkressource hat.

#### Sicherheit durch Gruppenrichtlinien

Um die Verwaltung von Tausenden von Objekten zu vereinfachen bzw. zu standardisieren, können Gruppenrichtlinien (GPOs) für diese Objekte erstellt werden. Gruppenrichtlinien regeln außerdem die Sicherheit von Computern und Benutzern im AD. Ein Gruppenrichtlinienobjekt enthält ein umfassendes Profil von Sicherheitsberechtigungen und wird verwendet, um Gruppenrichtlinien für Benutzer und Computer in AD auf Standort-, Domänen- und OU-Ebene anzuwenden. Ein einzelnes Gruppenrichtlinien-Objekt kann bspw. auf alle Computer einer Organisationseinheit angewendet werden. Die Gruppenrichtlinie wird angewendet, wenn der jeweilige Computer gestartet wird und unabhängig von weiteren Neustarts in regelmäßigen Abständen aktualisiert, um eventuell erfolgte Änderungen zu übernehmen. Vererbung von GPOs innerhalb der AD-Struktur ist möglich und kann dem Administrator viel Arbeit abnehmen.

Durch GPOs können bestimmte Sicherheitsrichtlinien, welche für Benutzer gelten sollen, domänenweit festgelegt werden. Dies sind zum Beispiel Kennwortrichtlinien, Kontosperrungsrichtlinien oder Kerberos-Ticketrichtlinien.

### 2.4.2 Protokolle und Mechanismen zur Authentifizierung

Ein Verfahren der Authentifizierung und Autorisierung von Benutzern stellt das Grundprinzip im Windows-2000-Sicherheitskonzept dar: Benutzer melden sich am System an und werden erst anschließend zur Nutzung bestimmter Ressourcen berechtigt. Diese Strategie wird mit Hilfe einer Reihe von einzelnen Sicherheitsfunktionen umgesetzt, welche Vertraulichkeit, Integrität und Verfügbarkeit von Daten im Netzwerk gewährleisten. Da

die Sicherheitsfunktionen an verschiedenen Stellen im Netzwerk arbeiten, wird hierbei auch oft vom Konzept der verteilten Sicherheit gesprochen.

Authentifizierung ist die Sicherstellung der Identität eines Subjekts. Ein Rechner muss erstens die Identität einer Person oder des Clients überprüfen, dann aber auch sicherstellen, dass berechtigten Benutzern der Dienst nicht verweigert wird. Bei der Feststellung der Benutzeridentität führt Windows 2000 auf der Grundlage der dem Benutzer eingeräumten Rechte und der mit der Ressource verknüpften Berechtigungen eine Überprüfung der Zugriffssteuerung durch.

Die Benutzerauthentifizierung von Windows 2000 besteht aus zwei unterschiedlichen Anmeldungsarten:

- Die interaktive Anmeldung:
  Von einem Benutzer, der auf die Windows 2000-Umgebung zugreifen möchte, wird verlangt, dass er sich entweder mit einem Domänen- oder einem lokalen Computerkonto anmeldet. Die Anmeldedaten entsprechen geheimen Daten, die der Benutzer oder Administrator für das lokale oder das Domänenkonto gesetzt hat.

- Die Netzwerkauthentifizierung:
  Hier wird bestimmten Netzwerkdiensten ein Identitätsnachweis bereitgestellt. Für Benutzer von lokalen Computerkonten stellt die Netzwerkauthentifizierung einen manuellen Prozess dar, der für jeden angeforderten Netzwerkdienst wiederholt werden muss. Domänenkonten-Benutzer profitieren von der sogenannten Single-Sign-On-Anmeldung, bei der sich ein Benutzer nur einmal im System anmelden muss. Benutzername und Passwort werden demnach kein zweites Mal abgefragt. Bei allen Zugriffen auf sich im Netzwerk befindliche Ressourcen erfolgt danach eine für den Benutzer selbst völlig transparente Authentifizierung und Zugriffssteuerung.

Bei den einfachen Authentifizierungsmechanismen vor Windows 2000 wurde der Zugriff auf Computer und Benutzerkonten mit Hilfe von Kennwörtern gesteuert. Dies bestätigte jedoch lediglich die Identität des Benutzers, indem er ein nur ihm bekanntes Geheimnis mitteilte. Geteilte Geheimnisse stellen zwar noch immer die Grundlage der Authentifizierung dar, jedoch haben sich die zum Schutz dieser Geheimnisse eingesetzten Mechanismen und Protokolle im Vergleich zu Windwos NT gewandelt. Die Sicherheitssysteme Windows 2000 unterstützen wirksame Authentifizierungsprotokolle, so zum Beispiel „Kerberos v5“, ein Mechanismus zur interaktiven Anmeldung und Netzwerkauthentifizierung, welche an Stelle der noch in Windows NT 4.0 verwendeten NTLM-Authentifizierung tritt. Um die Abwärtskompatibilität zu Windows NT-Rechnern zu gewährleisten, wird NTLM allerdings weiterhin in Windows 2000 unterstützt. Dies stellt jedoch auf der anderen Seite ein nicht zu verachtendes Sicherheitsrisiko dar, dessen sich Administratoren gewiss sein müssen.

**Das Kerberos Protokoll**

Wie im Abschnitt Windows NT 4.0 bereits kurz angesprochen, gelangt die NTLM-Authentifizierung bei größeren Netzwerken sehr schnell an ihre Leistungsgrenzen. Zudem kann die Verschlüsselung der geheimen Daten heutzutage nicht mehr als sicher angesehen werden. Daher wurde nach einer effizienteren Methode für die Authentifizierung gesucht. Fündig wurde Microsoft im UNIX-Segment: Dort wurde der Quellcode als kostenloser Open Source Code offeriert, welchem sich Microsoft bediente. Man machte einen weit verbreiteten Standard, der zudem nichts kostete, zum neuen Hauptauthentifizierungsprotokoll von Windows 2000. Die Auswirkungen dieser Aktion auf das Image von Microsoft sollen hier nicht weiter diskutiert werden...

Kerberos ist ein Protokoll, das zur Authentifizierung von Benutzern in „unsicheren“ Netzwerken entwickelt wurde. Microsoft benutzt in Windows 2000 die Kerberos-Version 5.5. Ein Vorteil von Kerberos liegt auf der Hand: es ermöglicht Windows 2000 eine Verteilung der Sicherheitsaufgaben in einem heterogenen Netzwerk über Windows hinaus. Das heisst, prinzipiell könnte ein UNIX-Rechner die Authentifikation in einer Windows-Domäne übernehmen oder auch umgekehrt, dass die Konten von UNIX-Anwendern unter Windows 2000 verwaltet werden könnten. Weiterhin ist die verwendete Verschlüsselung weitaus sicherer als beispielsweise die von NTLM. Je nach verwendetem Algorithmus können die übergebenen Kennwörter mit 56 oder 128 Bit verschlüsselt werden. Hierbei wird das Prinizp der symmetrischen Verschlüsselung genutzt, was bedeutet, dass ein Schlüssel sowohl zur Verschlüsselung als auch zur Entschlüsselung verwendet wird.

Der von Kerberos genutzte Mechanismus zur Authentifizierung von Konten und Berechtigungen entspricht den im RFC 1510 [1] beschriebenen und standardisierten Verfahrensweisen (RFC = Request for Comments). Hierbei teilt Kerberos die Authentifikation auf drei Systeme auf, dem Client, dem Server und dem Key Distribution Center (KDC). Das KDC fungiert als vertrauenswürdiger Vermittler und ist für die Schlüsselverteilung zuständig. Für den Zugriff auf Ressourcen jeglicher Art löst der Client beim KDC ein sogenanntes Ticket und präsentiert dieses beim Server. Die Rolle eines Key Distribution Centers kann bei Windows 2000 im Prinzip jeder Domänencontroller zusätzlich übernehmen, so dass jeder DC in der Lage ist, einem Client ein Ticket für den Zugriff auf Serverressourcen auszustellen. Der Prozess der Ticketvergabe sieht wie folgt aus:

> Meldet sich ein Benutzer an einem Windows 2000 Computer in der Domäne an, findet eine Authentifizierung am KDC, also an einem Domänencontroller, statt (siehe Abbildung 2.4). Dabei wird dem Anwender ein sogenanntes Ticket Granting Ticket (TGT) ausgestellt, das im Folgenden für die weitere Kommunikation mit dem KDC erforderlich ist. Möchte der Benutzter nun auf eine Netzwerkressource zugreifen, legt der Client dem Domänencontroller sein TGT vor und fordert ein Dienstticket an. Es enthält eine Bestätigung des DCs, dass der Anwender authentifiziert wurde und über welche Zugriffsberechtigungen er verfügt. Der Client legt das Dienstticket dann dem Server vor, der die Ressource bereit hält. Greift der Anwender später nochmals auf diese Ressource zu, ist keine Kontaktierung des Domänencontrollers mehr nötig, da das Dienstticket während der Arbeitssitzung seine Gültigkeit behält. Die vom

Authentifizierungsverkehr verusachte Netzwerklast wird durch diese Variante im Vergleich zu NTLM drastisch reduziert. Über lokale Sicherheitsrichtlinien oder besser noch domänenweite Gruppenrichtlinien lassen sich Eigenschaften wie z.B. die „Haltbarkeitsdauer“ eines Tickets bestimmen. Wenn mit Kerberos eine Verbindung zu Computern aus anderen Domänen hergestellt werden soll, so muss vorher eine Vertrauensstellung zwischen ihnen eingerichtet worden sein (siehe Abbildung 2.5).

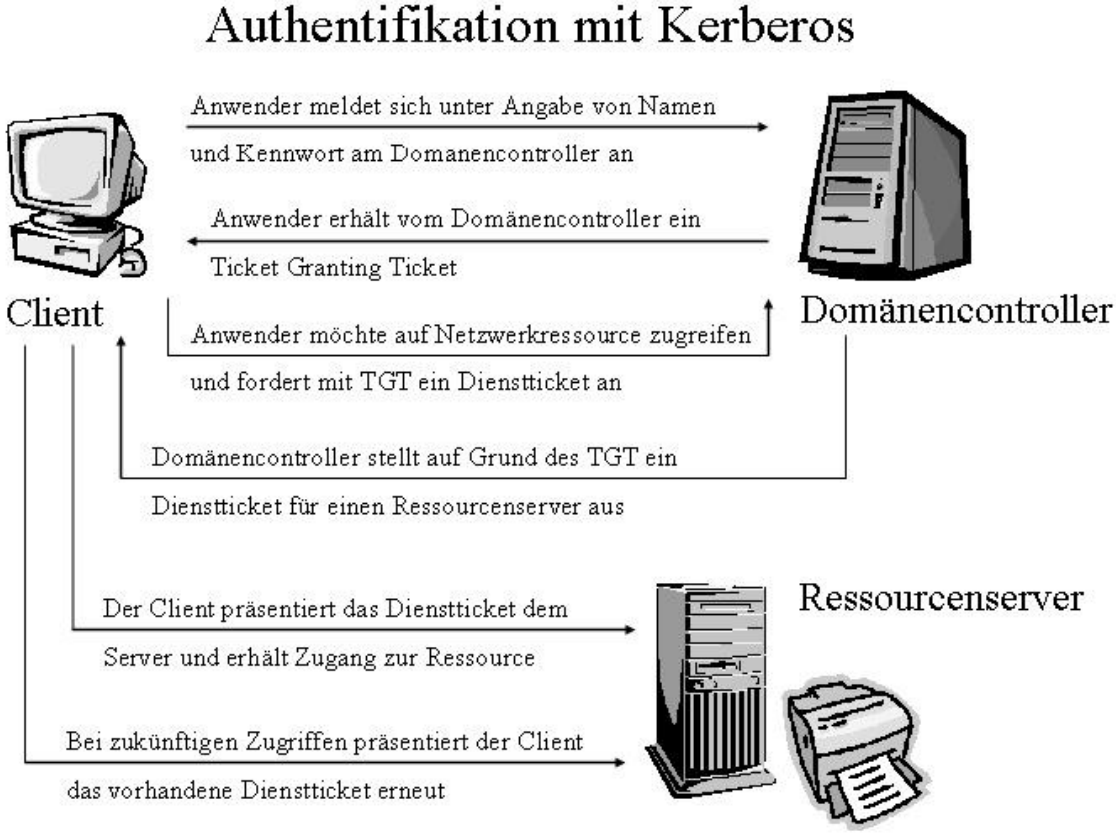


Abbildung 2.4: domänen-interne Authentifikation mit Kerberos [Quelle: „Windows Sicherheit“, Addison Wesley Verlag]

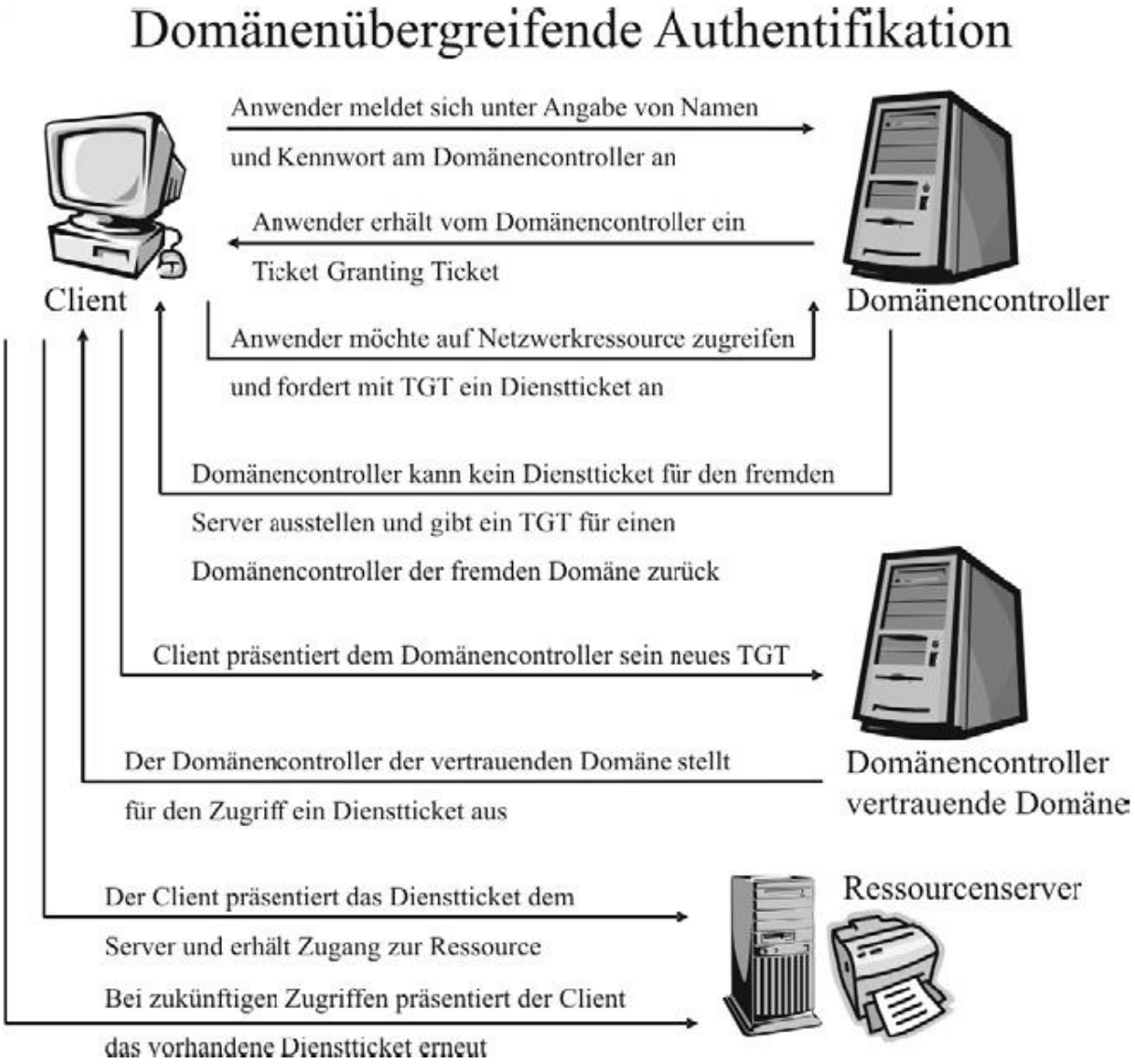


Abbildung 2.5: domänen-übergreifende Authentifikation mit Kerberos [Quelle: „Windows Sicherheit“, Addison Wesley Verlag]

Das Kerberos-Protokoll unterstützt standardmäßig nur das symmetrische Verschlüsselungsverfahren. Um aber eine Smartcard-Authentifizierung zu ermöglichen, erweiterte Microsoft das Protokoll um das Public-Key-Verschlüsselungsverfahren.

### Authentifizierung über Smartcard

Eine andere Variante zur Authentifizierung von Personen sind die sogenannten Smartcards. Statt Benutzername und Passwort zu präsentieren, legt der Benutzer hier eine Smartcard in einen dafür vorgesehenes Lesegerät ein. Sie sieht aus wie eine Telefonkarte und verfügt über einen kleinen Speicher, der mithilfe eines Schreib-Lese-Geräts das Zertifikat des Anwenders speichert (synchrone Chipkarten). Es gibt zudem Smartcards, die über einen eigenen Prozessor und ein Betriebssystem verfügen (asynchrone Chipkarten). So wird u.A. erreicht, dass der private Schlüssel nie preisgegeben werden muss. Nachdem

die zu verschlüsselnden Daten an die Smartcard übergeben wurden, führt diese dann die Verschlüsselung durch. Voraussetzung dafür ist natürlich die richtige PIN.

Die Basis der Smartcard-Authentifizierung besteht aus einem Zertifikat, welches von einer Zertifizierungsstelle erworben werden kann. Wird eine Authentifikation gegenüber einer Netzwerkressource notwendig, so wird dieses Zertifikat von dem sogenannten EAP/TLS-Protokoll (Extensible Authentication Protocol / Transport Layer Security) verwendet. Ebenso ist es mit Windows 2000 möglich, einem Anwender Zugriff auf einen Client einer Domäne zu gewähren. Hierzu wird ein Zertifikat verwendet, das auf Basis des X.509-Standards erstellt wurde und mit anderen Authentifizierungsdiensten zusammenarbeitet. Nachdem man eine Smartcard eingelegt hat, muss man im sogenannten GINA-Fenster (Graphical Identfication and Authentication) die PIN-Nummer eingeben. Die LSA (Local Security Authority) des entsprechenden Rechners überprüft nun die Richtigkeit der PIN. Bei korrekter Eingabe liest die LSA das gespeicherte Zertifikat von der Smartcard ein und übergibt es mithilfe des Kerberos-Dienstes an einen Domänencontroller der Domäne. Es erfolgt nun eine Verifizierung des privaten Schlüssels. Wird im Active Directory ein passendes Konto gefunden, erhält der Anwender, wie im letzten Abschnitt beschrieben, sein TGT (siehe Abbildung 2.6). Dieses wird aus Sicherheitsgründen mit dem öffentlichen Schlüssel des übergebenen Zertifikats, gemäß dem Prinzip der Public-Key-Verschlüsselung, codiert, so dass nur das System mit der Smartcard das TGT lesen kann. Der öffentliche Schlüssel hierzu wird vorher dem KDC zugesandt. Nun kann der Anwender wie gewohnt auf die gewünschte Ressource zugreifen. Anmerkung: obige Ablaufbeschreibung bezog sich auf eine Smartcard, die nur über einen Speicher verfügt (synchron).

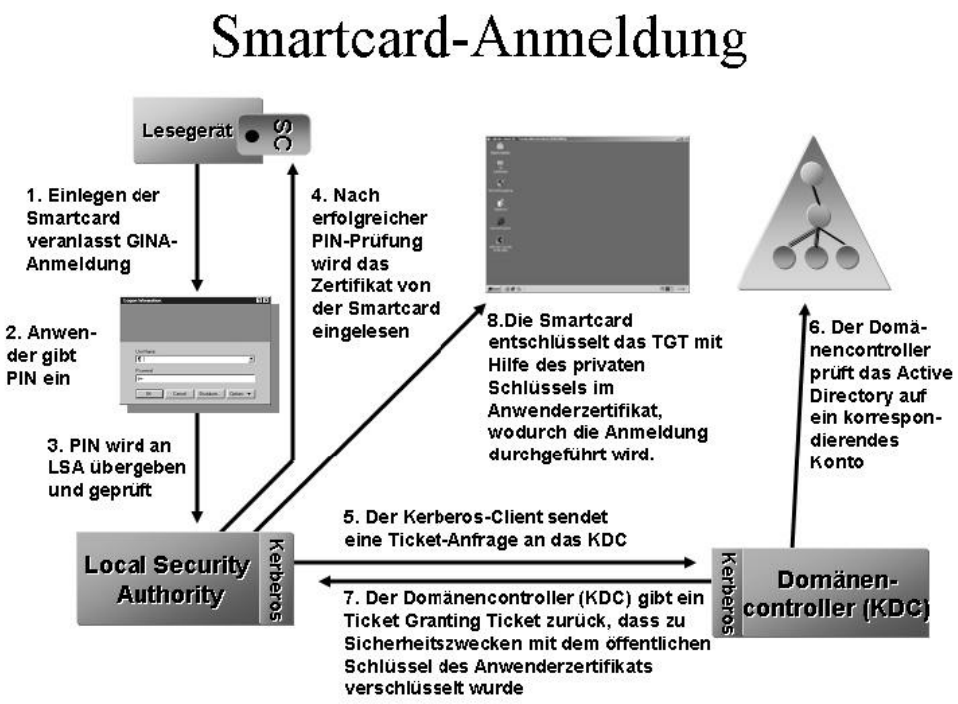


Abbildung 2.6: Smartcard-Authentifizierung [Quelle: „Windows Sicherheit“, Addison Wesley Verlag]

Zwei Vorbereitungen sind notwendig um in einem Windows 2000-System mit Smartcards arbeiten zu können:

- eine Domäne:
  nur bei einem existierenden Active Directory kann eine entsprechende Authentifikation durchgeführt werden.
- eine Zertifizierungsstelle:
  Diese muss in der Lage sein, Benutzerzertifikate, Enrollment-Agent-Zertifikate und Smartcard-Zertifikate auszustellen. Im Wesentlichen müssen dazu auf einem Domänencontroller die Zertifikatsdienste eingerichtet werden, um diesen so zu einer Certification Authority zu machen.

## 2.5 Die nächste Stufe - Authentifikation in Longhorn

Der nach aktuellem Stand voraussichtlich Mitte 2006 erscheinende XP-Nachfolger Longhorn soll in puncto Sicherheit und Schutz vor Datenmissbrauch entscheidende Verbesserungen integrieren. Unter anderem soll im Rahmen eines neuen Identity Systems die Kommunikation im Netz mittels eindeutiger, digitaler Identitäten sicherer gemacht werden. Für die Erkennung von Identitäten werden sogenannte iCards (Information Cards) benutzt, welche sich auf dem eigenen Computer befinden und an Kommunikationspartner im Netzwerk verschickt werden. Das System merkt sich gleichzeitig, an welche Ressource bereits eine iCard verschickt wurde. Alle Kontakte werden dabei grundlegend als Identitäten betrachtet, direkt als Objekte in WinFS-Dateisystem gespeichert und mit zahlreichen Informationen verknüpft. Hierzu gehören beispielsweise die E-Mail-Adresse eines Benutzers sowie - falls von diesem freigegeben - weitere persönliche Angaben, wie zum Beispiel seine Postanschrift. Neben seinem Public-Key-Zertifikat können zusätzlich auch Verwendungsrichtlinien, welche die Weitergabe persönlicher Daten an Dritte regeln, in einem solchen Objekt gespeichert werden. All diese Informationen werden von Longhorn entsprechend als iCard gespeichert und im Kontakt-Explorer abgelegt.

Ein mögliches Anwendungsbeispiel:

> In einem Fallbeispiel, welches auf der PDC-Konferenz in Los Angeles vorgestellt wurde, greift die Internet-Verbindungs-Firewall (ICF) auf iCards zurück, um Remote-Benutzern den Zugriff auf freigebene Ordner des PCs zu gewähren: Nach der Übermittlung der iCard durch den externen Benutzer und der dadurch sichergestellten Identität kann für ihn eine Berechtigung zur Remote-Nutzung erstellt werden. Werden diese Bedingungen - iCard-Prüfung, Ordner-Freigabe, Berechtigung - nicht erfüllt, so wird selbst ein Ping auf den jeweiligen Ressourcen-PC von der Longhorn-ICF abgewiesen. Entweder ist dem System die vorgelegte iCard nicht bekannt oder es mangelt an einer entsprechenden Berechtigung für diesen Benutzer. So verfügt Longhorn neben Authentifizierung und Autorisierung über eine zusätzliche Schutzebene, welche die Angriffsfläche des Computers einfach aber wirkungsvoll reduziert.

Ein Anwendungsbeispiel im Active Directory:

> Ein potentielles Unternehmen A könnte seine iCard an ein anderes Unternehmen B übermitteln. Nachdem die Identität im Active Directory des B erfolgreich geprüft wurde, wird ein Organisationsobjekt für A erstellt. Wenn nun ein Benutzer aus A mit dem Unternehmen B Kontakt aufnimmt, wird dieser als vertrauenswürdig erkannt. Die Identität des fremden Benutzers wird mithilfe von sogenannten Trustingbridges, die den Schlüssel der iCard verwenden, verifiziert. Der dazu passende Trustingbridge-Server wird voraussichtlich mit dem Erscheinen der Longhorn-Server-Version auf den Markt kommen bzw. bereits darin enthalten sein.

Neben dem Identity System soll Longhorn noch mit weiteren, neuen Sicherheitsfeatures aufwarten, wie zum Beispiel der neuen Code Access Security (CAS) oder dem Next Generation Secure Computing Base (NGSCB), auf welche in dieser Arbeit jedoch nicht weiter eingegangen wird.

# Zusammenfassung

Die in dieser Seminararbeit vorgestellten Mechanismen zur Authentifizierung unter dem Betriebssystem Microsoft Windows tragen maßgeblich zur Daten- und Kommunikationssicherheit in einem Netzwerk, sowie auch auf Einzelplatzrechnern, bei. Obwohl Windows den Ruf hat, unsicher zu sein, bin ich zu dem Schluß gekommen, dass Microsoft diesen Ruf weitgehend mit den Versionen bis einschließlich Windows NT erlangt hat. Im Wesentlichen werden heute die Mechanismen verwendet, die – wie dargestellt – erstmals mit Windows 2000 veröffentlicht wurden. Unter der Voraussetzung eines gut konfigurierten und mit aktuellen Sicherheitspatches erweiterten Systems kann man - so denke ich - ein Windows-Betriebssystem durchaus als sicher gegen die gängigsten Angriffsmethoden einstufen. Mit dem Verzeichnisdienst Active Directory hat Microsoft zudem ein Mittel zur sicheren Verwaltung von beliebig großen Netzwerken, welches auf dem Markt, meiner Meinung nach, seines Gleichen sucht.

# Literaturverzeichnis


[1] RFC 1510: `ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1510.txt.pdf`, besucht im Mai 2005

[2] „Active Directory sicher installieren und zuverlässig betreiben – Handbuch für bewährte Methoden“; Cole, Kreitler, Steen, Vilcinskas; Microsoft TechNet; Microsoft Corporation, 2003

[3] „backUP - Magazin für Sicherheit: MS Windows NT 4.0“, Landeszentrum für Datenschutz Schleswig-Holstein, 2002

[4] „backUP - Magazin für Sicherheit: Windows 2000“, Landeszentrum für Datenschutz Schleswig-Holstein, 2003

[5] „Windows Sicherheit“, Addison-Wesley Verlag, 2001

[6] „Windows 2000 Server in 21 Tagen“, Markt und Technik Verlag, 2000

[7] „PC Professional Expert Edition - Security Guide - Windows Sicherheit Netzwerk“, Tierling, Mergard, 2004

[8] „Seminar IT-Sicherheit - Sicherheit in Windows 2000“, Feti Saliji und Hadi Modarres

[9] „Orientierungshilfe - Datensicherheit ber der Installation und beim Betrieb von Windows NT“, Der Bayerische Landesbeauftragte für den Datenschutz, 2004

[10] „Workshop Sicherheit im Netzwerk, Windows 2000“, Herdt Verlag, 2000

[11] „Active Directory für NT Umsteiger“, Microsoft TechNet, David Melanchton, 2005

# Kapitel 3

# Authentifizierung in UNIX - Betriebssystemen

*Marcus Höppe*

*Authentifizierung ist eine der Grundlagen der Sicherheit im Bereich der Informationstechnologie. Mit ihrer Hilfe soll sichergestellt werden, dass ein Nutzer, der sich an einem System anmeldet, auch wirklich der ist, für den er sich ausgibt. UNIX ist nun ein Betriebssystem, das von vornherein als Multiuser-Betriebssystem angelegt worden ist. Die Schwierigkeit liegt dabei in den vorhandenen Ressourcen. Damit das quasi-gleichzeitige Nutzen der Ressourcen durch mehrere Nutzer auch gewährleistet werden kann, muss eine entsprechende Nutzer-Authentifikation durchgeführt werden. Im weiteren Verlauf dieser Arbeit sollen nun die grundsätzlichen Methoden zur Benutzerauthentifizierung unter UNIX sowie auch weiterführende Mechanismen behandelt werden. Besonderes Augenmerk soll dabei auf den Verfahren liegen. Die zugrunde liegenden Algorithmen werden wegen des quelloffenen Betriebssystems nicht näher betrachtet. Die Arbeit geht deshalb zunächst kurz auf die Geschichte von UNIX bis in die heutige Zeit ein. Anschließend wird die „normale“ Benutzerauthentifizierung unter UNIX behandelt. Danach werden weitere Möglichkeiten der Benutzerauthentifizierung unter dem Gesichtspunkt der Authentifizierung über ein Netzwerk betrachtet. Es wird dabei auf das Pluggable Authentication Module (PAM), das Simple Authentication and Security Layer (SASL) und das Lightweight Directory Access Protocol (LDAP) besonders eingegangen.*

## Inhaltsverzeichnis

## 3.1 Entwicklung von UNIX

UNIX ist ein Betriebssystem, das mittlerweile auf eine über 30jährige Geschichte zurückblicken kann. Es ist im Laufe der Zeit ständig weiterentwickelt worden und schlägt heute als eines der wesentlichen Betriebssysteme für PC's zu Buche. So lang die Geschichte von UNIX ist, so vielfältig ist sie auch. Im Folgenden soll nun die Geschichte von UNIX ein wenig näher behandelt werden.

### 3.1.1 Geschichtlicher Überblick

Die Geschichte von UNIX beginnt im Jahre 1969. Ken Thompson arbeitete zu dieser Zeit bei Bell Laboratories daran, ein neues Betriebssystem zu entwerfen. Ziel dieses Entwurfs war es, ein Betriebssystem zu entwerfen, das es ermöglicht, mit mehreren Nutzern gleichzeitig und im Dialog mit dem System arbeiten zu können (vgl. [2], Kap. 1.2). Bei herkömmlichen Betriebssystemen zu dieser Zeit wurde immernoch mit einem Operator gearbeitet, der die Programme auf Lochkarten kodiert dem Rechner eingab und die entsprechenden Resultate von einem Drucker holte (vgl. [3], Kap. 1.2.2). Diesen Vorgang verdeutlicht die folgende Abbildung (vgl. [2], Kap. 1.2). Thompson entwarf nun UNIX. Zunächst imple-

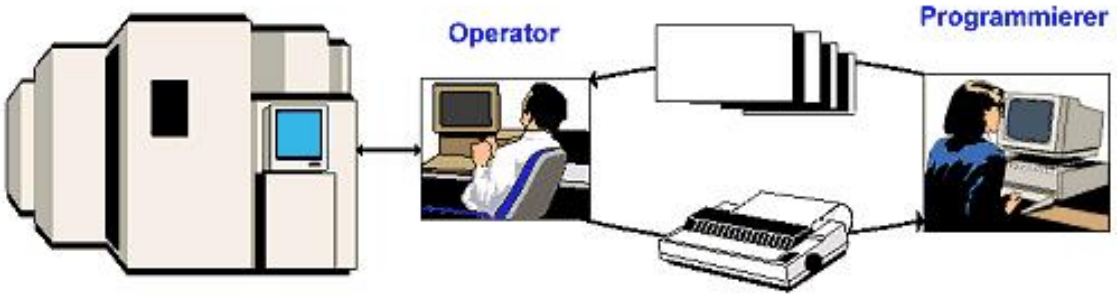


Abbildung 3.1: Batch-System

mentierte er UNIX in Assembler auf einer DEC PDP-7, einem Minicomputer. Ein anderer Forscher der Bell Labs, Brian Kernighan, nannte das System daraufhin scherzhaft UNICS (UNiplexed Information and Computing Service)[1] (vgl. [3], Kap. 10.1.1). Wenn auch nur scherzhaft gemeint, gelangte das Betriebssystem so zu seinem Namen. Nach dem Erfolg des Einsatzes auf der PDP-7, bei der UNIX seine Funktionalität unter Beweis gestellt hatte, sollte es auch auf neueren Rechnern eingesetzt werden. Damit nun UNIX nicht für den Einsatz auf jedem neuen Rechner in der Maschinensprache desjenigen Rechners neu implementiert werden mußte, benötigte man eine Sprache, mit der man eine systemunabhängige Implementierung erreichen konnte. Thompson entwickelte zu diesem Zweck die Sprache B[2]. Es gab aber auch mit B noch Probleme. Der Compiler erzeugte *threaded* Code, der dann interpretiert werden mußte, anstatt Maschinencode (vgl. [4], Kap. 1.1). Dennis

[1] MULTICS (MULTiplexed Information and Computing Service) war der Vorläufer von UNIX, entwickelt am M.I.T. und bei Bell Labs

[2] B war Nachfolger von BCPL (Basic Combined Programming Language)

Ritchie entwickelte aus B die Programmiersprache C und schrieb dafür einen Compiler, der Maschinencode erzeugte. Mit Hilfe von C wurde UNIX seitdem weiterentwickelt. Zunächst machte man UNIX portabel, sodass es auch auf anderen Rechnern lief, als auf der PDP-11. Die meisten Nutzer der PDP-11 zu dieser Zeit waren größere Unternehmen und vor allem Universitäten. Dadurch, dass das Standardbetriebssystem der PDP-11 recht unbeliebt war, verbreitete sich UNIX schnell (vgl. [3], Kap. 10.1.2). Der Sourcecode des Systems war frei zugänglich und so wurde damit auch entsprechend experimentiert. Im Laufe der Zeit entwickelten sich zwei konkurrierende Lager heraus, die beide ihr eigenes UNIX produzierten. AT&T entwickelten das System V und die Universität von Kalifornien in Berkeley (UCB) entwickelte BSD[3]. Berkeley wurde dabei vom Verteidigungsministerium finanziell unterstützt. Im Laufe der Zeit hatten viele Firmen UNIX Lizenzen erworben und produzierten ihre eigenen UNIX-Systeme. Dadurch war man bald an einem Punkt der Inkompatibilität der Systeme untereinander angekommen. Um eine variantenübergreifende Nutzbarkeit der entwickelten Software zu gewährleisten, wurde mit POSIX[4] ein Standard von der IEEE geschaffen, der standardisierte Bibliotheksprozeduren vorschreibt, die alle UNIX-Varianten implementieren müssen (vgl. [3], Kap. 10.1.4). Die nachstehende Abbildung soll die unterschiedlichen Entwicklungen von UNIX im Laufe der Zeit verdeutlichen (vgl. [2], Kap. 1.2).

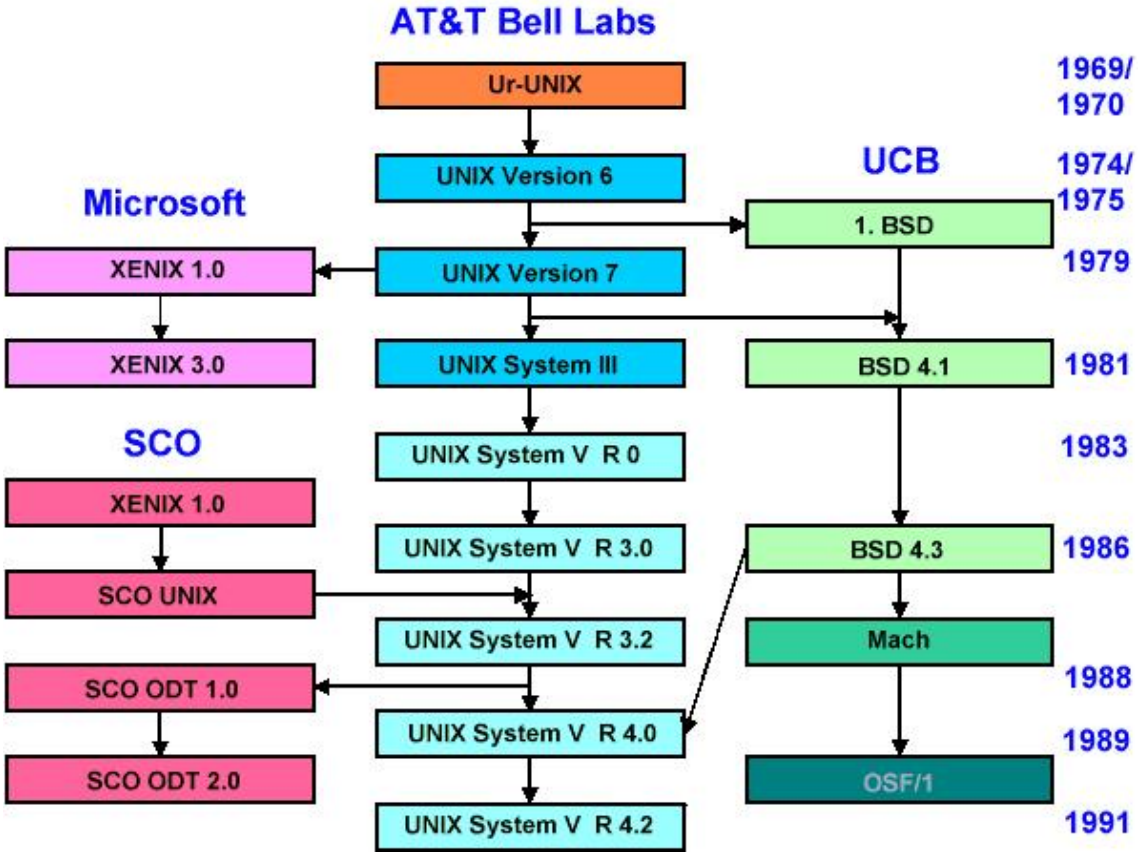


Abbildung 3.2: UNIX Entwicklung

[3]Berkeley Software Distribution
[4]Portable Operating System IX, wobei IX an UNIX erinnern soll

### 3.1.2 UNIX heute

Die heutige Entwicklung von UNIX ist durch eine große Anzahl weiterer Verzweigungen im Entwicklungsbaum von UNIX gekennzeichnet. Eine wesentliche Entwicklung in diesem Bereich ist Linux, das zwar eine Neuimplementierung darstellt, aber doch auf UNIX[5] aufbaut. Weitere Entwicklungen von UNIX in den letzten Jahren sind vor allem die Entstehung von NetBSD (aus BSD hervorgegangen), FreeBSD (von NetBSD abgespalten) und OpenBSD (von NetBSD abgespalten). Damit ist aber die Mannigfaltigkeit von UNIX noch nicht am Ende. Einige große Unternehmen, so z.B. Sun, HP, Apple,... haben eigene Varianten von UNIX für ihre Systeme hervorgebracht und betreuen und entwickeln diese Varianten stetig weiter.

## 3.2 Authentifizierung unter UNIX

Im nun folgenden Abschnitt soll die Funktionsweise der Benutzer-Authentifizierung in UNIX-Systemen erläutert werden. Dabei können durch die Vielfalt und teilweise auch die Altersunterschiede der vorhandenen UNIX Varianten eventuell kleine Unterschiede bestehen. Darauf wird an entsprechender Stelle im Text hingewiesen und teilweise eingegangen.

### 3.2.1 Grundlagen

Die Grundlagen der Authentifizierung unter UNIX lassen sich relativ kurz zusammenfassen. Die zur Nutzer-Authentifizierung notwendigen Daten werden unter UNIX in drei Dateien gespeichert. Diese drei Dateien liegen in der Dateistruktur unter dem Verzeichnis /etc. Abbildung 3.3 zeigt eine mögliche UNIX Dateistruktur (vgl. [2], Kap. 1.5.2). Die

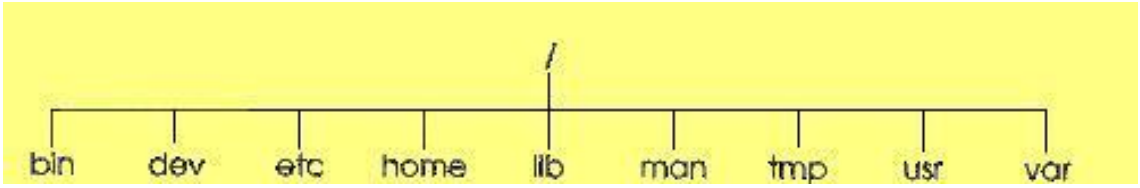


Abbildung 3.3: UNIX Dateistruktur

beiden entscheidenden Dateien in diesem Verzeichnis sind /etc/passwd und /etc/shadow. Die angesprochene dritte Datei ist die Datei /etc/group, auf die nur kurz eingegangen wird. Die Dateien sind ganz normale Textdateien, die folgende Aufgabe und Struktur haben:

[5]bzw. auf MINIX, einem kleinen Übungsbetriebssystem, das A.S. Tanenbaum für seine Studenten ursprünglich geschrieben hatte

- **/etc/passwd:** Diese Datei nimmt die Benutzerdaten und das Passwort[6] auf. Die Einträge in die Datei werden zeilenweise vorgenommen. Für jeden Benutzer wird also eine Zeile angelegt. Die Zeilen bestehen aus sieben Feldern, die durch Doppelpunkte voneinander getrennt sind. Die Zeilenlänge muß kleiner als 512 Zeichen sein, die einzelnen Felder sind dabei in ihrer Länge beliebig. Der Aufbau einer Zeile sieht wie folgt aus (die folgenden Daten beruhen auf [2], Kap. 1.7.2):
  **Login-Name : Passwort : UID : GID : Kommentar : **
  **Home-Directory : Programm**
  Dabei bedeuten die verschiedenen Felder:
  - **Login-Name:** der dem Betriebssystem bekannte Name eines Users
  - **Passwort:** das Passwort des Users in verschlüsselter Form (früher, siehe unten) bzw. ein 'x' um anzuzeigen, dass das Passwort in /etc/shadow steht (in neueren Systemen)
  - **UID:** die User ID des Benutzers, z.B. 0 = Superuser, normale User > 99
  - **GID:** die Gruppenzugehörigkeit des Users
  - **Kommentar:** in diesem Feld kann z.B. der reale, vollständige Name des Users stehen und/oder dessen Abteilung, etc.
  - **Home-Directory:** hier steht das Startverzeichnis für den jeweiligen User, z.B. /home/karl
  - **Programm:** hier steht das Programm, dass nach erfolgreicher Benutzeranmeldung automatisch gestartet werden soll, meist eine Shell.
- **/etc/shadow:** In dieser Datei steht das verschlüsselte Passwort. Der Grund dafür ist folgender: auf /etc/passwd haben alle User Lesezugriff. Das bedeutet, dass das Passwort eines jeden Users für jeden User sichtbar ist. Zwar ist es verschlüsselt, aber wenn man den Verschlüsselungsalgorithmus kennt[7], kann man sich das Passwort sorglos in verschlüsselter Form kopieren und eine Brute-Force Angriff auf das Passwort starten, um es im Klartext zu erhalten. Auch wäre der Kopiervorgang nicht nachvollziehbar, weil er in keinem Logfile auftauchen würde - schließlich stehen die Leserechte jedem User zu! Um diese Sicherheitslücke zu schließen, hat man das Passwort schließlich in die Datei /etc/shadow geschrieben und verweist in /etc/passwd nur noch auf ein vorhandenes Passwort. Auf /etc/shadow hat nur der Superuser Lese- und Schreibrechte. Die Datei enthält die Informationen ebenfalls in Zeilen, wobei jede Zeile in neun Felder unterteilt ist (vgl. [2], Kap. 1.7.4):
  **Name : Passwort : letzte Änderung : Min : Max : Vorwarnzeit : **
  **Inaktiv : Verfall : Kennzeichen**
  Die Bedeutung der verschiedenen Felder ist:
  - **Name:** derselbe Benutzername wie in /etc/passwd
  - **Passwort:** das verschlüsselte Passwort
  - **letzte Änderung:** die Zeit der letzten Änderung des Passwortes in Tagen (gezählt ab dem 01.01.1970)[8]

[6] bei früheren Systemen, heute werden /etc/passwd und /etc/shadow benutzt um Nutzerdaten und Passwort zu trennen

[7] die ersten zwei Zeichen des Passwortes geben den Verschlüsselungsalgorithmus an

[8] dem offiziellen Entstehungsdatum von UNIX

  - **Min:** die minimale Gültigkeitsdauer des Passwortes in Tagen (vorher ist kein Ändern möglich)
  - **Max:** die maximale Gültigkeitsdauer des Passwortes in Tagen (das Passwort muss vorher geändert werden)
  - **Vorwarnzeit:** die Zeit in Tagen vor dem Ablauf der Gültigkeit des Passwortes, in der der User auf den bevorstehenden Passwortwechsel aufmerksam gemacht wird
  - **Inaktiv:** die Anzahl an Tagen, die der User seinen Account unbenutzt lassen darf
  - **Verfall:** das absolute Datum, an dem die Nutzung des Accounts gesperrt wird
  - **Kennzeichen:** derzeit noch ohne Verwendung, mit 0 vorbelegt.

- **/etc/group:** Diese Datei speichert die Gruppenzugehörigkeit der Nutzer ab. Sie kann von allen Nutzern gelesen, aber nur vom Superuser beschrieben werden (vgl. [2], Kap. 1.7.3). Diese Datei besteht aus Zeilen zu je vier Feldern:
  **Gruppenname : Passwort : GID : Benutzernamen**
  Hinter den Einträgen verbirgt sich dabei:
  - **Gruppenname:** der Name der jeweiligen Gruppe von Nutzern
  - **Passwort:** dieses Feld bleibt leer
  - **GID:** in diesem Feld steht die Gruppen-ID
  - **Benutzernamen:** hier stehen sämtliche Nutzer, die dieser Gruppe angehören.

### 3.2.2 Login - Vorgang

Nachdem die drei Dateien, die für die grundlegende Nutzerauthentifizierung unter UNIX verantwortlich sind, beschrieben wurden, soll nun kurz auf den Login-Vorgang selbst eingegangen werden. Das Login wird mit Hilfe des sogenannten getty[9] Prozesses realisiert. Für die unterschiedlichen E/A - Geräte sind verschiedene getty's zuständig. Aufgaben von getty's können z.B. ermöglichen einer Login-Sitzung, einstellen der Parameter eines Terminals, ermöglichen von Einwahlverbindungen, etc. (vgl. [5], Die Login-Verwaltung) sein. So wird mit Hilfe der gettys auch die Nutzerkennungs- und Passworteingabe vorgenommen. Anschließend startet das getty dann den login-Prozess, der das eingegebene Nutzerkennung/Passwort Paar verifiziert.

## 3.3 Erweiterte Authentifizierungsverfahren

Zur Authentifizierung ist allerdings mit /etc/passwd, /etc/shadow und getty noch lange nicht alles gesagt. Im Laufe der Zeit sind weitere Möglichkeiten der Authentifizierung

[9] „Get us a TTY“, TTY = Teletypewriter, die frühere Eingabekonsole am Terminal eines Mainframe Rechners

entwickelt worden, um dem Nutzer die Anmeldung am System zu erleichtern oder/und für eine flexiblere Authentifizierung sorgen zu können. Im Folgenden soll nun eine Auswahl dieser erweiterten Verfahren behandelt werden. Die Verfahren setzen dabei an unterschiedlichen Punkten an. Sie reichen vom modularen Ansatz auf Applikationsebene über Kommunikationsprotokolle bis hin zu Schichten im Kommunikationsprotokollstapel.

### 3.3.1 PAM

Der erste Ansatz einer erweiterten Authentifizierung basiert auf einem modularen Konzept. Dabei soll im Weiteren gezeigt werden, welche Möglichkeiten und Vorteile dieser Ansatz bietet und es sollen einige zusätzliche Informationen zu PAM gegeben werden.

#### Entwicklung von PAM

PAM steht für Pluggable Authentication Module. Es stellt ein API[10] für authentifizierungsbezogene Dienste (vgl. [7], Kap. 17.2) dar. PAM wurde entwickelt und spezifizert von Vipin Samar und Charlie Lai von Sun Microsystems im Jahre 1995 (vgl. [6]). Der Grund für die Entwicklung dieses Frameworks war, dass die sogenannten System Entry Services[11] bei jedem neuen Authentifizierungsverfahren das verwendet werden soll, neu angepasst werden mussten um die Funktionsfähigkeit gewährleisten zu können. Das stellte meist einen hohen Aufwand dar und sollte deshalb verändert werden. Als Ergebnis dieser Überlegungen entstand PAM und ermöglichte es jetzt, verschiedenste Authentifizierungsmechanismen zu verwenden ohne die Login-Dienste verändern zu müssen. Der große Vorteil von PAM besteht also darin, dass die Umgebungsvariablen erhalten bleiben, auch wenn man von einer Authentifizierungsart zu einer anderen wechselt. Ein anderer Vorteil ist, dass vernetzte, heterogene Strukturen nebeneinander existieren können, die alle unterschiedliche Authentifizierungsverfahren nutzen, ohne sich gegenseitig zu behindern (vgl. [6]). Nachdem Sun PAM für ihr Betriebssystem Solaris entwickelt hatte, wurde es 1995 durch die Open Software Foundation (OSF) im Standard 86.0 veröffentlicht (vgl. [8], [9]). Danach wurde PAM schrittweise auf weitere UNIX- und Linux-Plattformen portiert. Im Zusammenhang mit PAM ist auch der X/Open-Standard XSSO zum Thema des Single Sign-On entstanden.

#### Struktur von PAM

Um den vorgenannten Forderungen gerecht zu werden und die Vorteile zu nutzen, trennt PAM die verschiedenen Anwendungen von den Authentifizierungsmechanismen (vgl. [8]). Die Abkürzung PAM steht dabei für die Grundgedanken dieser Lösung. *Module* sagt aus, dass es sich um einen modularen Ansatz handelt, der auch in der Softwareentwicklung allgemein eine große Rolle für die Wiederverwendbarkeit und die Anwendungsunabhängigkeit von Programmteilen spielt. *Authentication* sagt aus, dass es sich um Authentifizierungsarten handeln soll, die modular eingebunden werden können. *Pluggable* schließlich

[10] Application Program Interface
[11] z.B. login, rlogin, telnet

bedeutet, dass man es mit einem hochgradig konfigurierbaren Ansatz zu tun hat. Das PAM-Framework besteht aus shared libraries (vgl. [8]). Die Module sind ebenfalls shared libraries und stellen einen bestimmten Authentifizierungsmechanismus dar. Die oben angesprochene Trennung von Applikationen und Mechanismen verdeutlicht Abbildung 3.4 (vgl. [8]). Bei Abbildung 3.4 handelt es sich um eine Darstellung des sogenannten

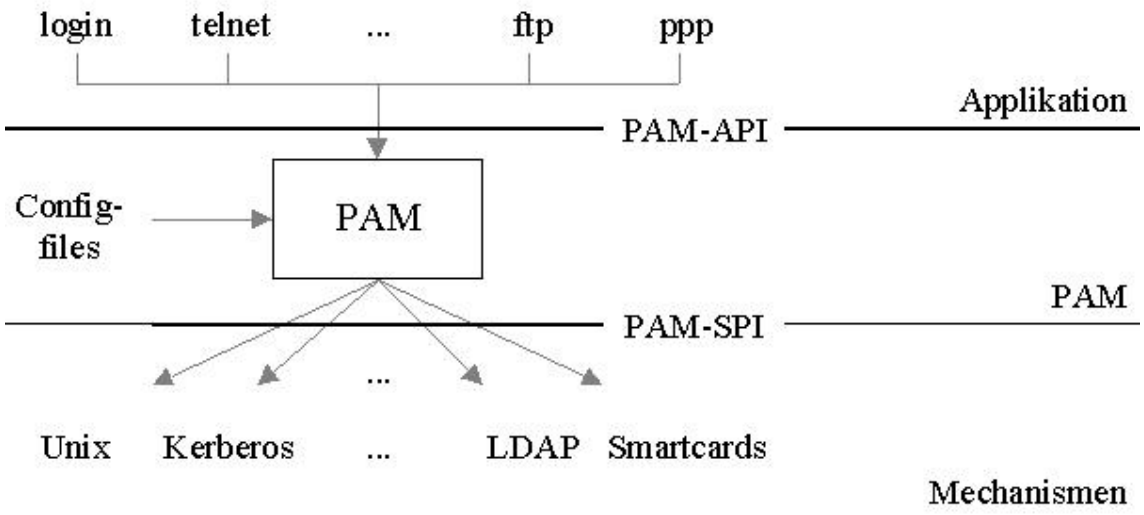


Abbildung 3.4: PAM Struktur

Authentication Management vom PAM. Wie man an der Abbildung erkennen kann, sind die Applikationen (oben: login, telnet,...) und die Authentifizierungsmechanismen (unten: Unix, Kerberos,...) getrennt und damit unabhängig voneinander. Mit Hilfe der in der Mitte der Abbildung dargestellten Schnittstelle (PAM) und einer entsprechenden Konfiguration (Config-files) ist es dann möglich, einen bestimmten Authentifizierungsmechanismus an eine bestimmte Applikation zu binden. Mit PAM ist es auch möglich, die verschiedenen Authentifizierungsmechanismen zu kombinieren (Module-Stacking). Außerdem bietet PAM zusätzlich Account Management, Session Management und Password Management. Diese Management-Funktionen sind ebenfalls in Form von Modulen in PAM integriert (vgl. [8]). Die Verbindung zwischen einer bestimmten Applikation und einem entsprechenden Mechanismus wird über genau definierte Interfaces hergestellt. Interfaces existieren für administrative Zwecke, für die Kommunikation zwischen Applikation und Modul, die Kommunikation zwischen Nutzer und Modul und die Kommunikation zwischen Modulen untereinander. Wie kann man nun PAM entsprechend konfigurieren? Die Konfiguration wird mit Hilfe der Datei pam.conf vorgenommen. In dieser Datei finden sich die Einträge, aufgebaut aus 5 Feldern. Diese 5 Felder sind der Reihe nach (vgl. [6]):

- **Service:** In diesem Feld wird die entsprechende Applikation (System Entry Service) eingetragen, z.B. login, rlogin,... . Anstelle einer dieser Applikationen kann auch der Wert other stehen. Dieser wird angegeben wenn alle anderen, nicht in pam.conf spezifizierten Dienste, gemeint sein sollen oder, falls alle Services dieselbe Konfiguration nutzen.

- **Module_type:** Dieses Feld beschreibt den Typ des PAM - Moduls. Als Möglichkeiten stehen auth, account, session oder password zur Verfügung. Dies sind genau die Management-Funktionalitäten.
- **Control_flag:** Hierbei wird das Verhalten beim Kombinieren von Modulen (Module-Stacking) beschrieben. Die dafür möglichen Werte sind requisite, required, sufficient oder optional.
- **Module_path:** Bei diesem Feld handelt es sich um die Angabe des Ortes, an dem sich das Modul befindet. Es ist möglich, einen default-Pfad anzugeben, um die Einstellungen so zu vereinfachen. PAM lädt das Modul nach Bedarf um die entsprechende Funktion einzubinden.
- **Field_options:** An dieser Stelle werden mögliche Optionen an das Modul weitergereicht.

Die folgende Abbildung verdeutlicht den Aufbau der PAM-Konfigurationsdatei (vgl. [6]). Es existiert nun, wie bereits vorher angesprochen, die Möglichkeit des Module-Stacking.

PAM Configuration File (pam.conf) with Different Modules

| Service | Module_type | Control_flag | Module_path | Options |
|---|---|---|---|---|
| login | auth | required | pam_unix_auth.so | nowarn |
| login | session | required | pam_unix_session.so | |
| login | account | required | pam_unix_account.so | |
| login | password | required | pam_unix_passwd.so | |
| ftp | auth | required | pam_skey_auth.so | debug |
| ftp | session | required | pam_unix_session.so | |

Abbildung 3.5: PAM Konfiguration

Dies soll nun folgend noch genauer erläutert werden. Das Konzept des Module-Stacking rührt aus folgendem Problem her: man nehme ein heterogen aufgebautes Netz von Rechnern. In einem solchen System besteht oft die Schwierigkeit, die unterschiedlichsten Authentifizierungsverfahren zu integrieren. Das Problem dabei ist, dass ein Nutzer das spezifische Authentifizierungsverfahren für jeden Rechner, an dem er sich anmelden will, und dazu auch die entsprechenden Kommandos kennen muss (vgl. [6]). Diese Tatsache spiegelt nicht gerade die pure Nutzerfreundlichkeit eines Systems wider. Um dieses Hindernis zu überwinden, entstand das Module-Stacking. Damit sollte es möglich sein, diese unterschiedlichen Anmeldeverfahren für den Nutzer transparent zu gestalten und den Vorgang zu vereinfachen. Die Abbildung 3.6 zeigt einen solchen Stack für den login-Prozess (vgl. [6]). Wie in der Abbildung zu sehen, ist dieser Stack nur für die Authentifizierungsfunktion im PAM-Framework eingerichtet, nicht aber z.B. für Session- oder Account-Management (vgl. [6]). Die bei der PAM Konfiguration weiter oben angesprochenen Controlflags kommen in diesem Fall zum tragen. Falls einer, oder mehrere Mechanismen einen Fehler auslösen, entscheidet das Controlflag welcher Fehler an die Anwendung weitergereicht wird.

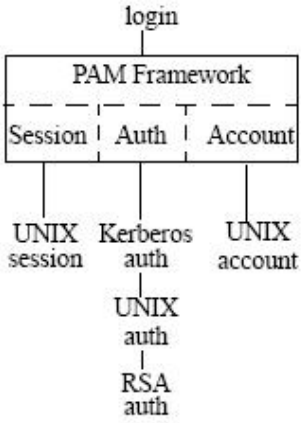


Abbildung 3.6: PAM Stack

Der Stack und damit die verschiedenen, kombinierten Verfahren zur Authentifizierung sollen für den Nutzer transparent bleiben. Die möglichen Conrolflags sind (vgl. [6]):

- **required:** Hierbei ist die Authentifizierung mit diesem Modul unbedingt erforderlich. Ein Fehler im Modul wird nach dem Aufrufen aller anderen Module auf dem Stack an den Aufrufer zurückgegeben. Damit die Funktion ein Ergebnis an die Applikation liefern kann, müssen alle Module, die dieses Flag nutzen, fehlerfrei durchlaufen worden sein.
- **requisite:** Mit diesem Flag wird gesagt, dass die Authentifizierung mit diesem Modul ebenfalls unerläßlich ist. Bei einem Fehler wird jedoch der Fehler sofort an den Aufrufer zurückgegeben, ohne, dass noch weitere auf dem Stack folgende Module durchlaufen werden. Auch hier gilt, dass für einen erfolgreichen Durchlauf alle Module fehlerfrei durchlaufen sein müssen.
- **optional:** Es wird genutzt, wenn der Nutzer Zugriffsrechte hat, auch wenn die Authentifizierung mit Hilfe dieses speziellen Moduls fehlgeschlagen ist. Bei einem Fehler wird in der Reihe der weiteren Module fortgefahren und der Fehler ignoriert.
- **sufficient:** Dabei wird bei fehlerfreiem Durchlauf eine entsprechende Erfolgsmeldung direkt an die Applikation gesendet, ohne folgende Module zu beachten, selbst wenn sie required oder requisite sind. Im Fehlerfall wird genauso vorgegangen, wie bei optional.

Abbildung 3.7 verdeutlicht die Einstellungen in der Konfigurationsdatei analog zum Authentication-Management Beispiel von Abbildung 3.6 (vgl. [6]). In dieser Konfigurationsdatei wird also folgendes festgelegt: dem Authentication-Management wird ein Modul-Stack übergeben. Die einzelnen Module des Stacks haben unterschiedliche Controlflags. Die Abarbeitungsreihenfolge geht dabei von unten nach oben. Das RSA-Modul hat als Flag optional eingetragen. Die folgenden Module UNIX und Kerberos haben beide required als Flag gesetzt. Das bedeutet, dass selbst wenn die Authentifizierung per RSA nicht erfolgreich ist, sich der Nutzer immernoch am System authentifizieren kann. Dabei muss er jedoch die Module für UNIX- und Kerberos-Authentifizierung ohne Fehler passieren (vgl.

PAM Configuration File with Support for Multiple Authentication Modules

| Service | Module_type | Control_flag | Module_path | Options |
|---|---|---|---|---|
| login | auth | required | pam_kerb_auth.so | debug |
| login | auth | required | pam_unix_auth.so | use_mapped_pass |
| login | auth | optional | pam_rsa_auth.so | try_first_pass |

Abbildung 3.7: PAM Stack Konfiguration

[6]). Nachdem nun die Entstehung und die Funktionalität von PAM im Zusammenhang weiterführender Authentifizierungsverfahren erläutert wurde, soll jetzt auf zwei weitere Möglichkeiten der Authentifizierung eingegangen werden.

### 3.3.2 SASL

SASL steht für Simple Authentication and Security Layer. Es bietet eine Möglichkeit, verbindungsorientierte Kommunikationsprotokolle um Authentifizierungsunterstützung zu erweitern (vgl. [10]). Die Abbildung 3.8 zeigt die Authentifizierung mit SASL (über LDAP) und einen anschließenden Datentransfer. Ein Protokoll, das SASL nutzen möchte, muss da-

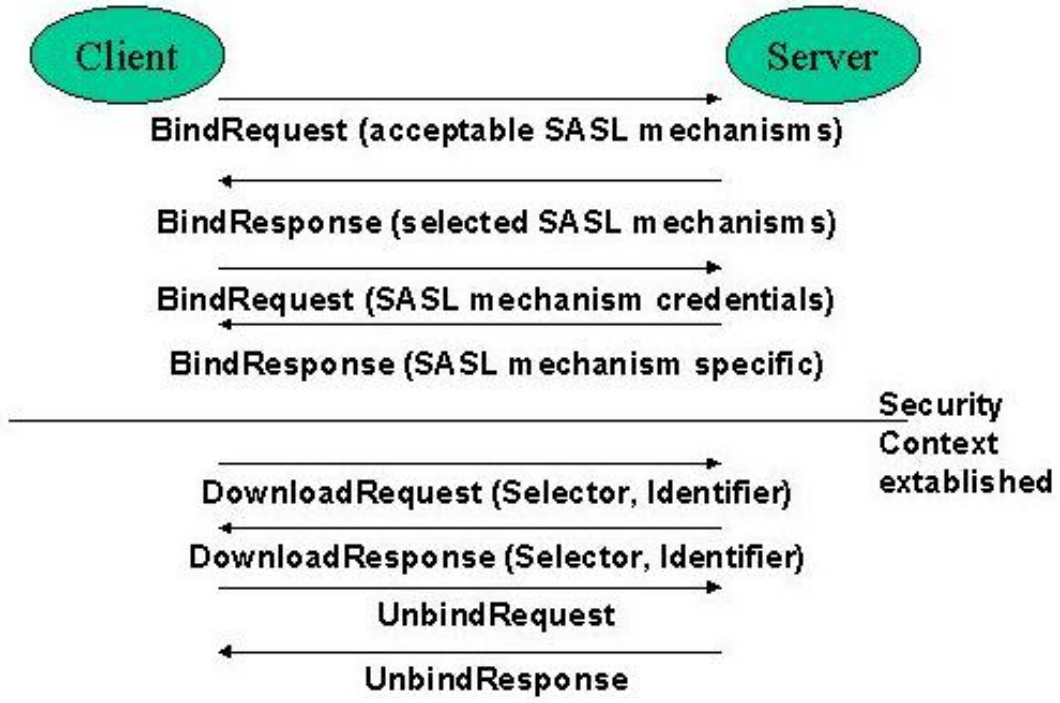


Abbildung 3.8: Authentifizierung mit SASL

bei über ein Kommando zur Identifizierung und Authentifizierung an einem Server verfügen. Das Kommando besitzt ein Argument, dass den zu verwendenden SASL-Mechanismus bezeichnet. SASL-Mechanismen werden mit Strings bezeichnet, die 1 bis 20 Zeichen lang

sein dürfen. Diese Mechanismen müssen bei der IANA (Internet Assigned Numbers Authority) registriert werden (vgl. [10]). Falls ein Server den entsprechenden Mechanismus unterstützt, initiiert er einen Datenaustausch zur Verhandlung des zu verwendenden Authentifizierungsprotokolls. Dieser Austausch besteht aus verschiedenen Challenges vom Server und entsprechenden Responses vom Client. Die Art und Anzahl hängt vom verwendeten Mechanismus ab. Sowohl bei Fehler als auch bei Erfolg des Challenge-Response Datenaustausches wird durch das Protokoll festgelegt, welche Reaktion erfolgt. Das gilt sowohl für die Server- als auch für die Clientseite. Während des Datenaustausches führt der Mechanismus die Authentifizierung durch, übermittelt die Identität des Aufrufers an den Server und vereinbart eine Mechanismus-spezifische Verschlüsselung der Daten. Wenn diese vereinbart worden ist, wird sie aktiv, sobald der letzte Challenge-Response Austausch stattgefunden hat. Alle folgenden Daten werden dann verschlüsselt zwischen Client und Server übertragen (vgl. [10]). SASL soll in Zukunft noch mehr Möglichkeiten bieten. Inzwischen wurde bei der IETF (vgl. [12]) ein neuer Internet Draft eingereicht (vgl. [11]), der den alten Standard von SASL und damit die alte Spezifikation RFC 2222 (vgl. [10]) ersetzen soll. Die Erweiterung von SASL soll dahin gehen, dass SASL auch in der Lage sein soll, Protokolle und Mechanismen zu trennen. SASL soll zu einem Framework erweitert werden, mit dem eine Konfiguration von Protokollen und Mechanismen frei möglich ist. Dabei soll SASL ein strukturiertes Interface zwischen den verschiedenen Protokollen und den unterschiedlichen Mechanismen zur Verfügung stellen. Ziel ist es, neue Protokolle mit bestehenden Mechanismen nutzen zu können und neue Mechanismen verwenden zu können, ohne Protokolle extra neu definieren zu müssen. Dieser weiterführende Ansatz ähnelt dem modularen Konzept von PAM. Die folgende Abbildung soll die zukünftig zu erwartende Struktur von SASL darstellen (vgl. [11], Kap. 2).

Nachdem in diesem Beispiel für weiterführende Authentifizierungsmöglichkeiten eine

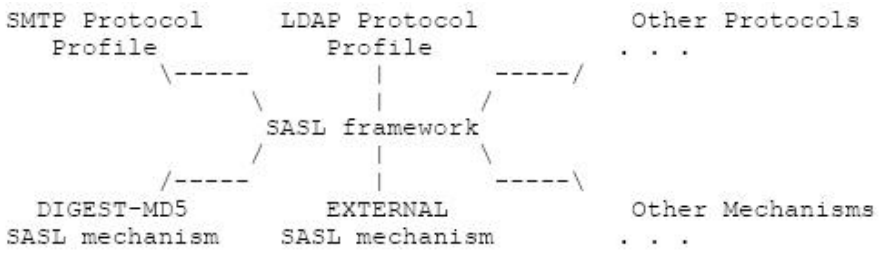


Abbildung 3.9: Konzeptionelles SASL Framework

Schicht behandelt wurde, so ist es im Folgenden ein Protokoll, auf das näher eingegangen werden soll.

### 3.3.3 LDAP

LDAP ist ein Netzwerkprotokoll, das für den Umgang mit Verzeichnisdiensten entwickelt worden ist. Es soll nun kurz die geschichtliche Entwicklung von LDAP beschrieben werden. Anschließend wird dann das Protokoll selbst genauer behandelt.

**Entwicklung von LDAP**

Die Geschichte hin zu LDAP reicht einige Jahre zurück. 1988 wurde von der ISO der X.500 Standard für Verzeichnisdienste als ISO 9594 verabschiedet. X.500 stellt eine Verzeichnisorganisation mit Hilfe hierarchisch gegliederter Namensräume zu Verfügung (vgl. [13]). Mit verschiedenen Suchfunktionen ausgestattet, stellt es einen sehr umfang- und funktionsreichen Standard dar. Zur Kommunikation zwischen Client und Server, die diese Funktionalitäten nutzen wollen, wurde DAP (Directory Access Protocol) erdacht. DAP ist ein OSI (Open Systems Interconnection)-Protokoll und setzt in den unteren Schichten des ISO/OSI Basisreferenzmodell (ISO 7498) die Protokolle der OSI-Familie voraus. DAP über TCP/IP war nicht vorgesehen. Die folgende Abbildung zeigt diesen Sachverhalt (vgl. [13]). Dadurch ist DAP auch sehr umfangreich in der Spezifikation und deshalb

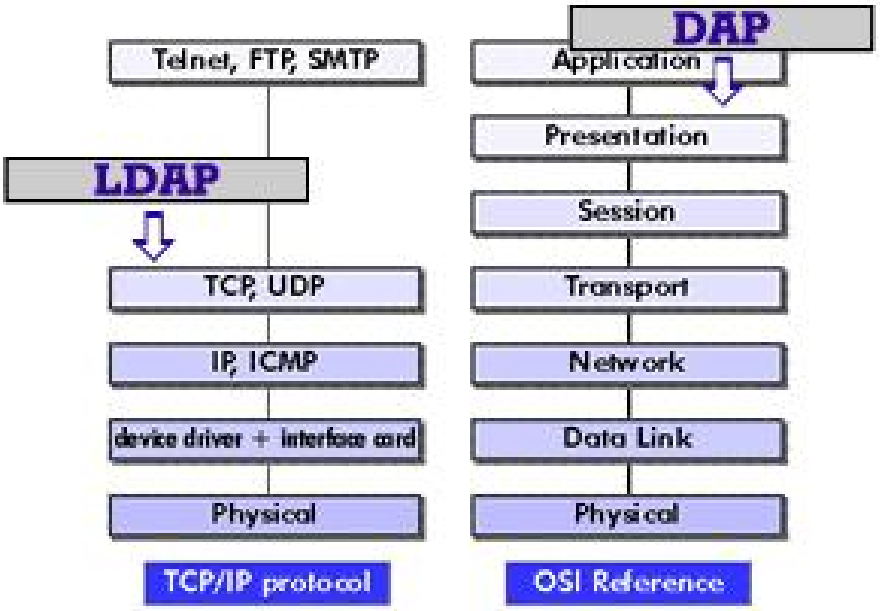


Abbildung 3.10: Vergleich LDAP und DAP

umständlich zu implementieren. Außerdem sah der X.500 Standard die Veröffentlichung einiger Verzeichnisdiensteinträge im Internet vor (vgl. [15], Kap. 8), womit Unternehmen, die ein Hauptanwendungsgebiet für Verzeichnisdienste darstellen, nur schwerlich einverstanden gewesen wären. Nun wurde durch eine gemeinschaftliche Entwicklung mehrerer Unternehmen eine „abgespeckte“ Version von DAP entwickelt - LDAP. LDAP setzt auf dem TCP/IP-Protokollstapel auf und bietet damit eine auf das Internet zugeschnittene Implementierung (siehe Abbildung 3.10), was es erheblich einfacher gestaltet als DAP. LDAP wurde 1995 in der Version 2 im RFC 1777 festgeschrieben, Version 3 folgte 1997 im RFC 2251 (vgl. [12]).

**Struktur von LDAP**

LDAP wird durch vier Modelle beschrieben (vgl. [15], Kap. 8, [13]):

- **Informationsmodell:** Es beschreibt, wie die Daten, die im Verzeichnis vorhanden sind, strukturiert sind.
- **Namensmodell:** Es beschreibt, wie die Daten adressiert werden.
- **Funktionsmodell:** Es beschreibt, mit welchen Funktionen auf dem Datenbestand gearbeitet werden kann.
- **Sicherheitsmodell:** Es beschreibt, wie Daten auf eine sichere Art und Weise gespeichert und vor nicht autorisiertem Zugriff geschützt werden.

Diese Modelle sollen jetzt behandelt werden. Die im Folgenden verwendeten Informationen und Bilder beruhen dabei auf [13].

**Das Informationsmodell:** Alle Daten zusammen, die in einem Verzeichnis zu finden sind, werden Directory Information Base (DIB) genannt. Die DIB besteht aus sogenannten Einträgen (engl. Entry). Ein Eintrag ist dabei eine Anzahl von Informationen über ein sogenanntes Objekt. Ein Objekt modelliert meist eine Sache (Gegenstand, Person,...) aus der Realität (vgl. [13]). Ein Alias bezeichnet einen Eintrag, die auf einen anderen Eintrag verweist. Einträge wiederum bestehen aus einer Anzahl von Attributen. Attribute besitzen einen Typ und einen oder mehrere Werte. Der Attributtyp wird syntaktisch festgelegt. Die Syntax bestimmt auch, welche Parameter bei Suchoperationen zum Tragen kommen (z.B. ignorieren der Groß-/Kleinschreibung). Attribute haben zwei Typen, mandatory (notwendig) und optional. Optionale Attribute können weggelassen werden. Es ist auch möglich, dass Attribute mehrfach in einem Eintrag vorkommen. Die nachstehende Abbildung zeigt die Organisation der DIB (vgl. [13]). Man kann die Struktur von LDAP auch im Stil

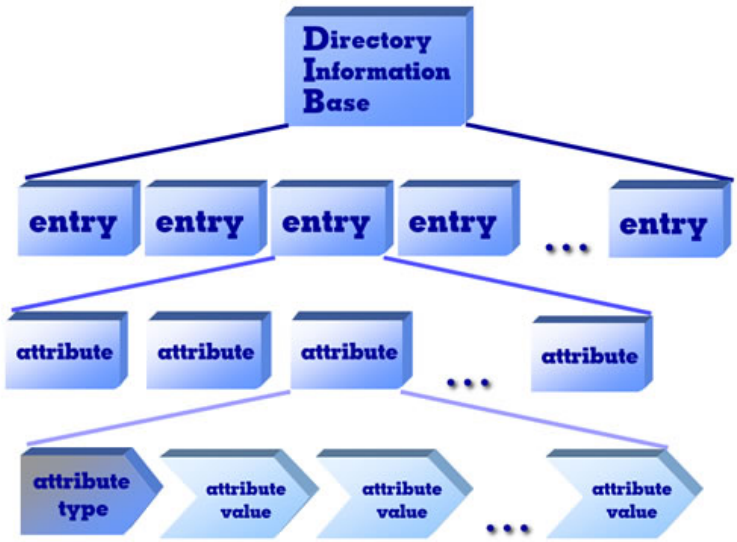


Abbildung 3.11: Aufbau der DIB

einer objektorientierten Programmiersprache auffassen. Danach entspräche jeder Eintrag

einem Objekt, also einer Instanz einer Klasse. Eine Klasse fasst Objekte gleichen Typs zusammen. Die Klasse vererbt ihre Attribute und es können weitere hinzukommen. Im RFC 2252 ist festgelegt, dass LDAP Objekt-Klassen entweder

- **abstrakt,**
- **strukturell** oder
- **helfend** (engl. auxiliary) sein können (vgl. [13]).

Eine **abstrakte Klasse** dient nur dazu, andere Klassen von ihr abzuleiten. Die abgeleiteten Klassen erben von dieser abstrakten Oberklasse die Attribute. Auf diese Art und Weise, kann man sich ohne viel Aufwand ein Template für eine Reihe abzuleitender Klassen schaffen (vgl. [13]). Abstrakte Klassen haben keine Einträge. **Strukturelle Klassen** sind dadurch gekennzeichnet, dass Einträge eine solche besitzen müssen. Ein Eintrag hat also immer eine strukturelle Oberklasse. Bei LDAP sind die meisten Klassen von dieser Art. **Helfende Klassen** schließlich werden benutzt um gleiche Attribute an verschiedene Einträge zu vergeben. Die objektorientierte Analogie dazu ist die, dass eine Klasse Methoden von mehr als einer Oberklasse erben kann, also mehr als eine Oberklasse besitzen kann. Einträge, die zu dieser helfenden Oberklasse gehören müssen auch noch eine weitere, strukturelle Oberklasse besitzen. In einem sogenannten Schema wird in einem Directory-Server eine zusammenfassende Beschreibung aller Objekt-Klassen, die verwendet werden, abgelegt (vgl. [13]). Dieses Schema sagt aus, welche Klassen es gibt, welche Attribute eine Klasse besitzt, welche davon sie besitzen darf, welche sie besitzen muss und in welcher Syntax sie beschrieben sind.

**Das Namensmodell:** Jede Eintrag hat einen eindeutigen Distinguished Name (DN). Dieser sagt aus, wo im Verzeichnisbaum sich der entsprechende Eintrag befindet. Einträge

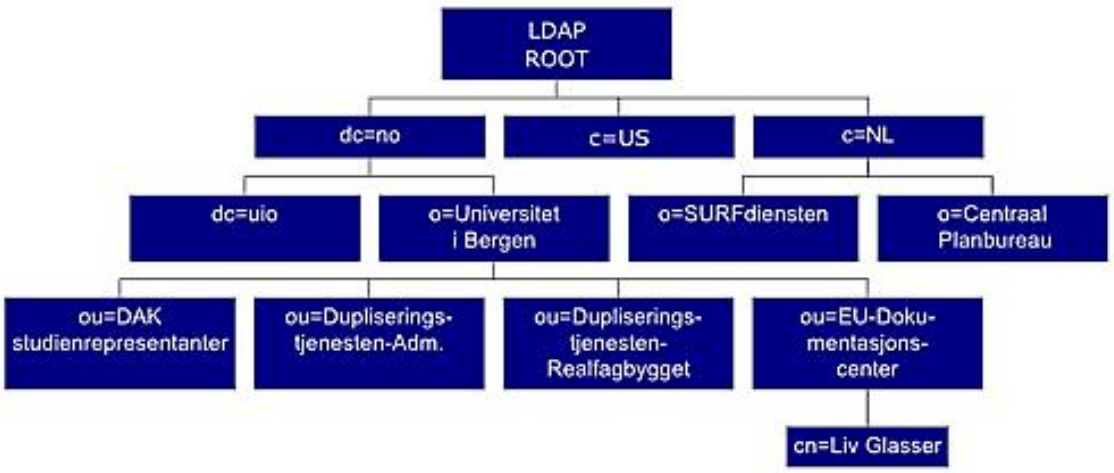


Abbildung 3.12: Ein Namespace

können Parent-, Peer- und Child-Einträge besitzen (vgl. [13]). Ein DN besteht aus Relative Distinguished Names (RDN). Der komplette Verzeichnisbaum ist im sogenannten Directory Information Tree (DIT) repräsentiert. Die weltweite Eindeutigkeit eines DN ist in LDAP nur im Zusammenhang mit der LDAP-URL[12] gegeben. Eine LDAP-URL setzt sich aus der URL des LDAP-Servers und des entsprechenden DN zusammen, z.B. ldap://ldap.canon.com/o=canon,c=us (vgl. [13]). Die Namen eines Directories stellen seinen Namespace dar (vgl. [13]). Die Abbildung 3.12 zeigt einen Namespace (vgl. [13]). Zu diesem Namespace existiert natürlich eine anwendungsbezogenere Sicht, die browserähnlich aufgebaut ist. In dieser Ansicht läßt sich dann genau erkennen, welche Attribute und welche Werte zu jedem Eintrag existieren. Die folgende Abbildung zeigt, wie ein solcher Eintrag im Einzelnen aussieht (vgl. [13]). Mit Hilfe der RDN's der Person Liv

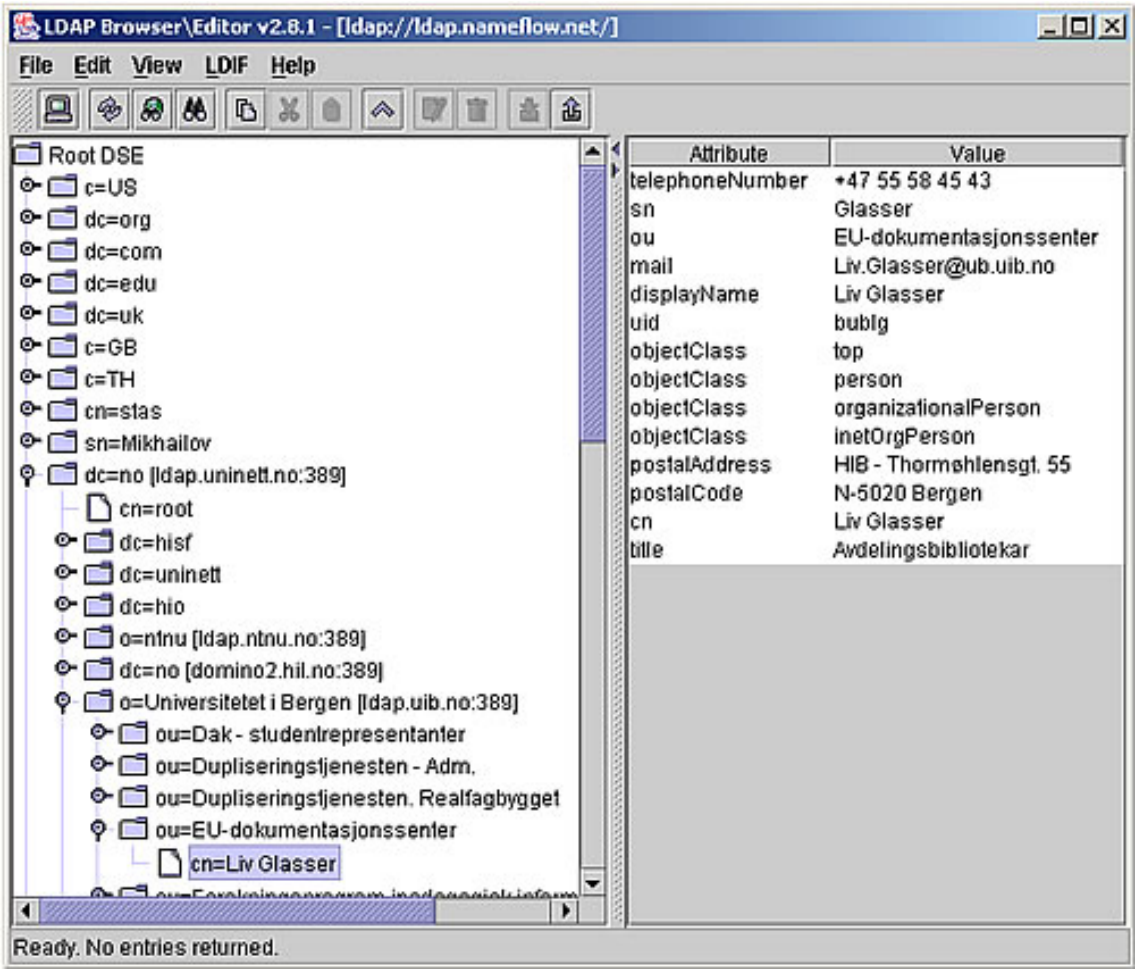


Abbildung 3.13: LDAP Browser

Glasser kann man ihre DN zusammensetzen. Die DN lautet: cn=Liv Glasser, ou=EU-dokumentasjonssenter, o=Universitetet i Bergen,c=NO. Damit läßt sich dieser Eintrag im übergeordneten LDAP-Server eindeutig bestimmen (vgl. [13]).

[12]Uniform Resource Locator

**Das Funktionsmodell:** LDAP stellt verschiedene Funktionen bereit, mit denen man auf die Verzeichniseinträge zugreifen kann. Im wesentlichen sind das: suchen, modifizieren, hinzufügen, löschen, modifizieren von DN, vergleichen, abbrechen und die für die Authentifizierung wichtige Funktion bind (vgl. [16]).

**Das Sicherheitsmodell:** Im Sicherheitsmodell wird beschrieben, wie der Verbindungsaufbau, der Informationsaustausch und die Authentifizierung über LDAP funktionieren. Speziell auf die Authentifizierung mit LDAP soll im folgenden noch genauer eingegangen werden.

### Authentifizierung mit LDAP

Von LDAPv2 zu LDAPv3 haben sich einige Dinge getan (vgl. [18], Kap. 1.5). LDAP ist erweiterbar geworden, die Referenz-Weiterleitung hat sich verändert, Unicode wurde als die Standard-Kodierung festgelegt und es sind wesentliche Maßnahmen zur Authentifizierung, Integritätsprüfung und gesicherten Übertragung getroffen worden. In diesem Zusammenhang taucht auch SASL wieder auf, das bereits im Kapitel 3.3.2 besprochen worden ist. Mit diesen Erweiterungen wurden Mechanismen erdacht, um SASL und Transport Layer Security (TLS) mit LDAP zu verwenden. Beispielsweise wird im RFC 2829 (vgl. [17]) beschrieben, welche Maßnahmen für Directories mit bestimmten Anforderungen zu treffen sind. Darunter sind auch Maßnahmen, die die Nutzung von SASL explizit vorschreiben. Die folgende Abbildung illustriert schematisch die Authentifizierung via LDAP und SASL (vgl. [19], Kap. 6). Die im Weiteren verwendeten Informationen stammen aus [19], Kap. 3.

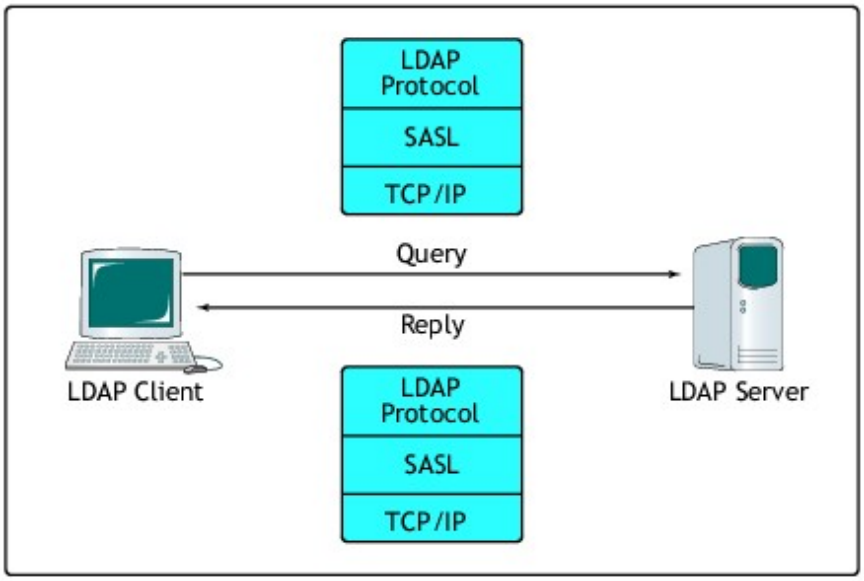


Abbildung 3.14: LDAP Authentifizierung mit SASL

Bei LDAP findet die Authentifizierung mit Hilfe der oben erwähnten Funktion bind statt. Mit bind kann sich ein Client an einem Server anmelden. Diese Operation ist aufgeteilt in bind request und bind response. In einem bind request sind die LDAP-Versionsnummer

und eine DN enthalten, die zur Identifizierung des Aufrufers dient (vgl. [19]). Außerdem enthalten ist ein Eintrag zur Authentifizierungsmethode und die Nutzdaten. Es werden zwei Methoden unterschieden: Simple bind und SASL bind. Beide Varianten sollen kurz erläutert werden.

- **Simple bind:** Bei dieser Methode wird zusätzlich zu den oben angesprochenen Inhalten des bind request ein Passwort mitgeschickt, das dann der Server vergleichen kann (vgl. [19]). Standardmäßig findet dabei keine Verschlüsselung der Verbindung statt. Um zu vermeiden dass das Passwort im Klartext übertragen wird, ist in diesem Fall eine Verschlüsselung, z.B. mit TLS, ratsam. Für einen möglichen anonymen Zugriff können DN und Passwort entfallen.
- **SASL bind:** Dabei werden die Möglichkeiten von SASL (siehe 3.3.2) genutzt. Es stehen verschiedene Modi zur Verfügung:
  - **PLAIN:** Diese einfache Authentifizierung mit Passwort ist nicht grundlegend anders als das Simple bind (vgl. [19]).
  - **KERBEROS_V4:** Es wird Kerberos Version 4 verwendet, um ein Kerberos Ticket auszustellen für ein Single Sign-On. Da Kerberos V4 Sicherheitslücken aufweist, sollte dieser Modus nicht mehr verwendet werden.
  - **GSSAPI:** Das GSSAPI (Generic Security Service Application Program Interface) stellt eine Schnittstelle dar, über die weitere Authentifizierungsmethoden zum Einsatz kommen können. So können zum Beispiel Kerberos V5 und auch X.509 als Modi verwendet werden. Auf diese beiden Modi soll hier nicht weiter eingegangen werden.
  - **DIGEST MD5:** Hier benutzt man einen Challenge-Response Ansatz, der darauf beruht, dass beim Client mit MD5 ein Hash von Passwort und empfangener Challenge gebildet und an den Server zurückgeschickt wird. Der Server kann dadurch feststellen, ob der Client das richtige Passwort besitzt.
  - **EXTERNAL:** Dabei wird eine bereits auf anderen Schichten bestehende Authentifizierung (z.B. TLS, IPsec, X.509, etc.) für die Anwendungsschicht verwendet.

Durch die Tatsache, dass die Authentifizierung mit SASL als eine fest integrierte Möglichkeit für LDAPv3 besteht, ergeben sich einige Vorteile. Zum einen braucht man sich nicht zusätzlich um Authentifizierungsmaßnahmen bemühen, sie stehen bereits zu Verfügung. Zum anderen ist durch die Flexibilität, mit der man SASL verwenden kann, auch das Integrieren weiterer Methoden, um die Sicherheit zu steigern (z.B. Verschlüsselung), ohne großen Aufwand möglich. Natürlich ist auch hier nicht alles Gold, was glänzt. LDAP ist durch seine zunehmende Verbreitung auch in zunehmendem Maß Angriffen ausgesetzt. Es wurden spezielle Tools entwickelt, die Brute-Force Angriffe auf LDAP-Verzeichniseinträge durchführen. Andererseits wiederum zeigt diese Tatsache nur umso deutlicher, dass die Entwicklung und Weiterentwicklung von Sicherheitsmaßnahmen einen hohen Stellenwert hat und auch weiterhin haben wird.

## 3.4 Zusammenfassung

Es ist bereits aus dem alltäglichen Umgang mit Rechnern, Servern und verschiedenen Netzwerken (Internet, Intranet, ...) bekannt, dass auch der verschiedenen Bedrohungen wegen, auf starke und verlässliche Sicherheitsmechanismen nicht verzichtet werden kann. Authentifizierung stellt im Spektrum dieser Mechanismen nur einen Aspekt dar. Wie aus den hier vorgestellten Authentifizierungsmethoden ersichtlich ist, wurde im Laufe der Zeit ein erheblicher Aufwand getrieben um die verschiedenen, existierenden Methoden zu ergänzen und flexibler zu gestalten. Durch das Auftreten immer neuer und weitreichender Authentifizierungsansätze (z.B. Biometrie) werden auch in Zukunft einfach zu erweiternde und zu konfigurierende Methoden und Anwendungen benötigt.

Im Zuge dieser Arbeit wurden nun verschiedene Authentifizierungsverfahren beleuchtet. Angefangen von der simplen Authentifizierung am UNIX-System selbst wurden mit PAM, SASL und LDAP drei weiterführende Möglichkeiten beschrieben, die man für eine Authentifizierung an UNIX-Systemen verwenden kann. Bei allen drei Verfahren wurden ihr struktureller Aufbau und die Möglichkeiten, die sie in puncto Authentifizierung bieten, behandelt. Mit PAM, SASL und LDAP ist im Rahmen dieser Arbeit jedoch nur auf eine kleine Auswahl der zur Verfügung stehenden Verfahren eingegangen worden. Der strukturelle Unterschied der drei Verfahren stellt dabei ein entscheidendes Kriterium dar. Es sollte möglichst ein weitgefächerter Bereich von Methoden und Anwendungen zur Authentifizierung abgedeckt werden.

# Literaturverzeichnis


[1] Plötner, J. Wendzel, S.: *Praxisbuch Netzwerk-Sicherheit, Galileo Press GmbH, 1.Auflage, Bonn 2005*

[2] Plate, J.: *Betriebssystem UNIX, http://www.netzmafia.de/skripten/unix/index.html, München 2005*

[3] Tanenbaum, A.S.: *Moderne Betriebssysteme, Pearson Studium, 2.überarbeitete Auflage, München 2003*

[4] Erlenkötter, H.: *C Programmieren von Anfang an, Rowohlt Taschenbuch Verlag, 8. Auflage, Reinbek 2004*

[5] Ermer, T. Meyer, M.: *Die Linuxfibel, http://www.linuxfibel.de*

[6] Samar, V. Lai, C.: *Making Login Services Independent of Authentication Technologies, http://www.sun.com/software/solaris/pam/pam.external.pdf*

[7] NetBSD.org: *Chapter 17. Pluggable Authentication Modules (PAM), http://www.netbsd.org/guide/en/chap-pam.html*

[8] Bähr, G.: *PAM - Pluggable Authentication Modules, http://wwwbs.informatik.htw-dresden.de/svortrag/ai96/Baehr/#absatz5_1*

[9] Samar, V. Schemers, R.: *Unified Login with Pluggable Authentication Modules (PAM), http://www.opengroup.org/tech/rfc/rfc86.0.html*

[10] Myers, J.: *Simple Authentication and Security Layer (SASL), http://www.ietf.org/rfc/rfc2222.txt*

[11] Melnikov, A. Zeilenga, K.: *Simple Authentication and Security Layer (SASL), Internet-Draft, http://www.ietf.org/internet-drafts/draft-ietf-sasl-rfc2222bis-11.txt*

[12] Internet Engineering Task Force (IETF): *http://www.ietf.org*

[13] Mitlinx.de: *LDAP, http://www.mitlinx.de/ldap/index.html*

[14] Marshall, B.: *Introduction to LDAP, http://www.quark.humbug.org.au/publications/ldap/ldap_tut.html*

[15] Hallberg, B.: *Netzwerke, IT-Studienausgabe, 1. Auflage, Bonn 2004*

[16] Wahl, M. Howes, T. Kille, S.: *Lightweight Directory Access Protocol (v3), http://www.ietf.org/rfc/rfc2251.txt*

[17] Wahl, M. Alvestrand, H. Hodges, J. Morgan, R.: *Authentication Methods for LDAP, http://www.ietf.org/rfc/rfc2829.txt*

[18] OpenLDAP: *Introduction to OpenLDAP Directory Services, http://www.openldap.org/doc/admin22/intro.html*

[19] Gietz, P.: *Chancen und Risiken LDAP-basierter zentraler Authentifizierungssysteme, http://www.daasi.de/pub/Chancen%20und%20Risiken%20durch%20LDAP-Authentifizierung.pdf*

# Kapitel 4

# Web Services Security

*Roman Goltz*

*Web Services sind ein relativ neuer Ansatz, verteilte Anwendungen zu realisieren. Da dieser Ansatz komplett von der Wirtschaft inspiriert wurde, liegt eine besondere Bedeutung auf der kommerziellen Nutzung. Wirtschaftsprozesse benötigen allerdings eine hohe Sicherheit, da teilweise immense Werte transferiert werden bzw. in Form von Aufträgen Kapital binden. Daher ist mit das wichtigste Kriterium für die Nutzbarkeit von Web Services ihre Fähigkeit, Sicherheitsanforderungen zu erfüllen.*
*In dieser Arbeit werden die Sicherheitanforderungen an Web Services untersucht, um anschließend Lösungen aus der Praxis zu diskutieren. Dazu werden zuerst einzelne Standards wie XML Encryption oder XML Signature diskutiert und später dann WS-Security untersucht, das einen übergreifenden Ansatz verfolgt.*
*Danach wird gezeigt wie neue Standards wie WS-Privacy, WS-Trust, WS-Police, WS-SecureConversation, WS-Authorization und WS-Federation auf WS-Security aufbauen und im Zusammenspiel die meisten Sicherheitsprobleme von Web Services lösen können.*

## 4.1 Web Services und deren Sicherheitanforderungen

In diesem Kapitel soll eine kurze Einführung in Web Services gegeben werden. Es wird keine Einführung in XML geben, allerdings wird kurz auf SOAP als Datenübertragungsmodell eingegangen. Am Ende werden Sicherheitsanforderungen an Web Services betrachtet, die die eigentliche Motivation zur Umsetzung von Web Service Security bieten.

### 4.1.1 Web Services

Web Services sind einige der wenigen Informationstechnologien, die nicht dem akademischen Sektor entsprungen sind, sondern aus der industriellen Praxis heraus entstanden sind. Zwei große Bedürfnisse der Wirtschaft waren dabei der Antrieb:

1. Die meisten Web-Anwendungen waren sogenannte „B2C“ (Business to Costumer) Anwendungen. Diese Dienste ermöglichen es Kunden, über das Web auf Angebote einer Firma zuzugreifen (z.B. Amazon). Wichtig hierbei ist natürlich ein entsprechendes Interface, um dem Kunden Kauf- bzw. Servicevorgänge so angenehm wie möglich zu gestalten. Sehr bald kam man auch auf die Idee „B2B“ Anwendungen zwischen Firmen zu realisieren, zum Beispiel um Nachbestellungen von Einzelteilen zu automatisieren. Dafür benötigt man keine schicken Benutzeroberflächen sondern effizente Möglichkeiten zur automatischen Abwicklung.
2. Ein weiterer wichtiger Ansatz war es, innerhalb eines Unternehmes alle vorhandenen Anwendungen so zu integrieren, dass sie wie eine homogenen Anwendung wirken und benutzt werden können, auch wenn die einzelnen Komponenten in verschiedenen Programmiersprachen geschrieben sind. Somit sollte auch die Effizenz gesteigert werden, denn mit diesem Ansatz brauch man nur noch ein Programm um die verschiedenen Anwendungen, die an einem Geschäftsvorgang beteiligt sind, zu steuern und kann somit Kosten einsparen.

Nachdem viele Firmen die Entwicklung vorangetrieben hatten, wurden Web Services (WS) vom World Wide Web Consortium (W3C) im Jahre 2003 standardisiert [3]:

„A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based message conveyed by Internet Protocols.“

Das bedeutet also, das Web Services Anwendungen sind, die über eine oder mehrere Schnittstellen von überall aus benutzt werden können. Web Services kann man leicht identifizieren, denn dazu nutzt man einfach das gängigue Format URI (Uniform Ressource Identifiers), das auch im WWW zur Seitenidentifizierung dient. Es gibt Anwendungen, die einem Dienstanbieter helfen, Web Services zu spezifizieren und dem Dienstnutzer helfen, diese zu finden. Die Kommunikation mit den Diensten findet in der Sprache XML statt, der Austausch der Nachrichten erfolgt über Standardprotokolle aus der Internetwelt.

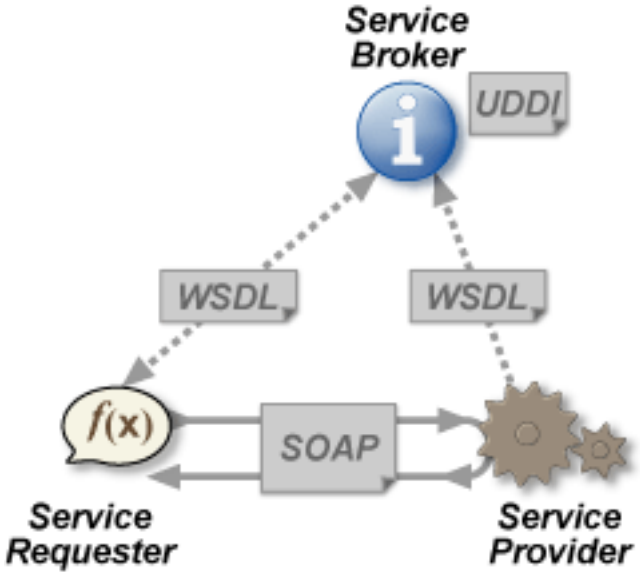


Abbildung 4.1: eine abstrakte Darstellung des Zusammenspiels verschiedener Parteien bei Web Services von [6]

### 4.1.2 SOAP

SOAP ist ein leichtgewichtiges Protokoll, dessen Zweck der Austausch strukturierter Informationen in einer dezentralen, verteilten Umgebung ist. Dazu wird basierend auf XML ein Rahmenwerk für den Datenaustausch geschaffen. Die so erstellten Nachrichten können dann über verschiedene Wege - also eine Auswahl verschiedener Transportprotokolle - übertragen werden. Das Nachrichtenformat ist komplett anwendungsabhängig, aber es ist natürlich möglich in diesen Nachrichten anwendungsspezifische Informationen zu transportieren.

Eine SOAP-Nachricht ist nach dem Head-Body Pattern modelliert. Im Head-Bereich der Nachricht werden die Metainformationen der Nachricht untergebracht. Diese können Informationen über das Routing der Nachricht, über eine eventuelle Verschlüsselung bzw. über die Zugehörigkeit zu einer Transaktion umfassen. Im Body der Nachricht sind, wie auch bei HTML, die Nutzdaten untergebracht. Diese Daten müssen vom Empfänger der Nachricht interpretiert werden, mögliche Zwischenstationen können diese auch ignorieren.

Als SOAP entworfen wurde war es erklärtes Ziel ein besonders einfaches, aber dafür erweiterbares Protokoll zu schaffen. In der Grundausstattung hat SOAP weder Lösungen für Sicherheit, noch für Verlässlichkeit oder Routing. Dafür bietet es Schnittstellen zur Erweiterung, die auch genutzt werden, wie wir später noch sehen werden.

### 4.1.3 Sicherheitsanforderungen an WS

In diesem Abschnitt soll geklärt werden, welche Sicherheitsanforderungen für Web Services bestehen. Diese werden dann später durch verschiedene Sicherheitsprotokolle erfüllt, wo-

mit diese Anforderungen die Motivation für die erstellung spezieller Sicherheitsprotokolle darstellen.

### Vertraulichkeit

Vertraulichkeit bedeutet, dass nur diejenigen eine Nachricht lesen können, für die sie auch bestimmt ist. Es heisst nicht, dass andere Personen die Nachricht nicht lesen können - sie könenn sie nur nicht verstehen. Für Web Services bedeutet Vertraulichkeit, dass niemand, der nicht die erforderlichen Rechte besitzt, nachvollziehen kann, welche Daten man an einen Web Service überträgt und welche Antworten man von diesem zurückbekommt.

### Authentizität

Authentizität ist erreicht, wenn man sich sicher sein kann, das der jeweilige Kommunikationspartner auch tatsächlich die Person ist, die man vermutet. Bei Web Services bedeutet Authentizität, dass der Dienstbringer sicher wissen kann, von wem ein Aufruf kommt und der Dienstnutzer nachvollziehen kann, von wem der Dienst erbracht wird.

### Integrität

Bei der Integrität geht es darum, sicherzustellen, dass eine Nachricht während des Transports nicht von Dritten verändert wird. Bei Web Services wäre eine mögliche Bedrohung, dass einerseits der Dienstaufruf verändert werden kann, so dass unter Umständen eine viel kompliziertere Anfrage an den Web Service entstehen kann. Genau so könne auch die Ergebnisnachrichten abgefangen werden.

### Nicht-Anfechtbarkeit

Viele geschäftliche Situationen erfordern es, dass zweifelsfrei nachgewiesen werden kann, dass ein bestimmtes Ereignis stattgefunden hat. Für Web Services geht es darum ein digitales Pendant zu finden, die Geschäftspartnern diese Sicherheit vermitteln kann.

### Verfügbarkeit und Zugangskontrolle

Hier geht es darum, dass Ressourcen im Netz auch verfügbar sein sollten, damit sie von Nutzen sind. Der beste Dienst bringt nichts, wenn man keine Antwort erwarten kann. Außerdem soll über Zugangskontrolle sichergestellt werden, dass nur derjenige einen Dienst nutzen kann, der auch die erforderlichen Zugriffsberechtigungen hat.

## 4.2 Security Standards von Web Services

Da Web Services laut ihrer Spezifikation auf bekannten Protokollen aufbauen sollen, hat man dies mit Sicherheitsprotokollen ebenso getan. Hier sollen nun diejenigen Protokolle aufgeführt werden, die später zum Aufbau neuer und komplexerer Sicherheitsprotokolle dienen, damit in späteren Abschnitten die Synergieeffekte erkannt werden können.

### 4.2.1 XML Encryption

Um die XML Nachrichten, die durch Web Services genutzt werden sicher zu machen, gibt es durch XML Encryption einen Standard, der das Verschlüsseln erleichtert. Durch XML Encryption wird ein XML Dokument in ein verschlüsseltes XML Dokument verwandelt. Das Wurzelelement heisst <EncryptedData> und dieses kann Kindelemente besitzen, die etwas über den Schlüsselalgorithmus aussagen.

Einer der Vorteile, die auch maßgeblich die Performance dieses System beeinflussen, sind die verschiedenen Granularitätsstufen. XML Encryption kennt davon 4 Stufen:

- Man kann ein komplettes Element verschlüsseln, also seinen Inhalt und seinen Namen. Damit wird nicht nur der Inhalt geschützt, sondern man kann auch nicht mehr erkennen, was für ein Element übertragen wurde.
- Man kann nur den Inhalt des Elements verschlüsseln.
- Man kann den Inhalt eines Elements verschlüsseln, der nur aus Text besteht, also keine Kinderelemente mehr besitzt.
- Natürlich kann man auch komplette XML-Dokumente verschlüsseln.

Dazu ein Beispiel. Folgende XML Datei zeigt ziemlich sensible Kreditkartendaten:

```
<?xml version='1.0' ?>
<PurchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
    <Quantity>12</Quantity>
  </Order>
  <Payment>
    <CardId>123654-8988889-9996874</CardId>
    <CardName>visa</CardName>
    <ValidDate>12-10-2004</CardName>
  </Payment>
</PurchaseOrder>
```

Daraus wird nach der Verschlüsselung der Kreditkartennummer folgende XML Datei:

```
<?xml version='1.0' ?>
<PurchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
    <Quantity>12</Quantity>
  </Order>
  <Payment>
    <CardId>
      <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Content'
                  xmlns='http://www.w3.org/2001/04/xmlenc#'>
        <CipherData>
          <CipherValue>A23B45C564587</CipherValue>
        </CipherData>
      </EncryptedData></CardId>
    <CardName>visa</CardName>
    <ValidDate>12-10-2004</CardName>
  </Payment>
</PurchaseOrder>
```

Zusammenfassend kann man sagen, das XML Encryption ein sehr nützliches Tool ist. Es kann sehr effizent gemacht werden, wenn man sehr feingranular arbeitet. Die aus Auswahl der Algorithmen ist frei wählbar und damit ist dieses Protokoll auch zukunftssicher.

### 4.2.2 XML Signature

XML Signature soll es möglich machen, XML Dokumente digital zu signieren und die Signatur ebenfalls in XML auszudrücken. Mit diesen Fähigkeiten lassen sich Authenzität, Integrität und Nicht-Anfechtbarkeit umsetzen. Der wichtigste Algorithmus ist SHA-1, weil er zum Signieren und insbesondere zur Bildung des Fingerabdrucks benötigt wird.

Das grundlegende Element, an dem eine Signatur erkennbar ist, ist <Signature>, alle weiteren Informationen sind als Kindelemente unter diesem Tag zu finden. Die wichtigsten Kindelemnete sind:

- <SignedInfo>: identifiziert und beschreibt die zu signierende Information
- <KeyInfo>: wird verwendet, um z.B. Zertifikate zu kodieren oder Referenzen auf Schlüsselspeicher abzulegen
- <SignatureValue>: Die eigentliche Unterschrift, die aus einem Schlüssel und dem zu signierenden Element erstellt wurde.
- <Object>: optionales Element, das signierte Elemnte enthalten kann

Die Signatur selbst kann in drei verschiedenen Beziehungen zu dem unterschriebenen Objekt stehen. Sie kann in das Objekt mit eingebettet sein, dann spricht man von einer Enveloped Signature. Andersherum kann auch das Objekt in die Signatur eingebettet sein - dann spricht man von einer Enveloping Signature. Zu guter letzt kann sich das Objekt an einem anderen Ort befinden, der über eine URI referenziert wird. In diesem Fall spricht man dann von einer Detached Signature.

## 4.3 WS-Security Initiative

Im letzten Abschnitt wurden mehrere verschiedenen Sicherheitsfeatures vorgestellt, die zur Verbesserung der Sicherheit von Web Services genutzt werden können. Allerdings ist insgesamt eine Vielzahl von Einzelstandards verfügbar, jedoch kein zusammenhängendes Sicherheitskonzept speziell für Web Services. Dieses Problem erkannten auch die großen Antreiber der Web-Services-Technologie und die Standardisierungsgremien. Im folgenden Abschnitt werden WS-Security und die darauf aufbauenden Protokolle erläutert und ihre Ziele näher definiert.

### 4.3.1 Begriffe für Web Services Sicherheit

Im Umfeld der Web Services Sicherheit werden einige Termini immer wieder benutzt, die an dieser Stelle erklärt werden sollen:

- **Security Token**: wird definiert als die Repräsentation von sicherheitsrelevanten Informationen (z.B.: Kerberos tickets, X.509 Zertifikat, Informationen von einer SIM-Karte im mobilem Umfeld, ein Username etc.)
- **Signed Security Token**: sind Security Tokens, die einen Satz von Claims enthalten, die durch kryptographische Verfahren durch den Aussteller bestätigt werden
- **Claims**: ein Claim ist eine Aussage über ein Subject von dem Subject selbst oder einer dritten Partei, das das Subject dem Claim zuordnet. Es werden keine Einschränkungen getroffen, was diese Claims beinhalten dürfen bzw. wie sie ausgedrückt werden.
- **Subject**: eines Security Tokens ist ein Bevollmächtigter (also z.B. eine Person, eine Anwendung oder eine Geschäftsstelle) auf den die Claims in dem Token zutreffen. Insbesondere muss ein Subject Informationen besitzen, welche es eindeutig als Besitzer des Tokens ausweisen.

### 4.3.2 Zusammenhang mit heutigen Sicherheitsmodellen

Die Überlegungen um Sicherheit von Web Services und die daraus entstandenen Protokolle sind kompatibel mit anderen heute genutzten Mechanismen zur Gewährleistung von

Authentizität, Datenintegrität und Nicht-Anfechtbarkeit. Es ist also möglich Web Service Security mit anderen Sicherheitsmodellen zusammen zu benutzen:

- **Transport Security** - Existierende Protokolle wie zum Beispiel die Secure Sockets (SSL/TLS) können einfache point-to-point Integrität und Nichtanfechtbarkeit bieten. Aufbauend darauf lassen sich zusammen mit WS-Security diese Eigenschaften auch auf end-to-end Verbindungen übertragen. Bei SSL/TLS werden immer komplette Nachrichten verschlüsselt. Für einfache Szenarien, wenn z.B. ein Client nur mit einem Server kommuniziert, mag das ausreichen. Allerdings werden auch viele Teile verschlüsselt, die nicht mal vertraulich sind. Wenn der Server viele Verbindungen hat, muss er relativ viel ver- und entschlüsseln. Ein weiterer Nachteil liegt in der SOAP-Architektur. Hier ist es vorgesehen, dass Zwischenpunkte auch in die Nachricht schauen können und zum Beispiel das Routing an eigene Vorschriften anpassen können. Dies ist nicht möglich, da Zwischenpunkte ja nicht über die erforderlichen Schlüssel verfügen. Daher benötigen wir feingranulare Verschlüsselung auf Nachrichtenebene. Daher wurden XML Encryption und XML Signature vorgestellt, die beide in WS-Security eingeflossen sind.
- **PKI** - Die Public Key Infrastruktur ermöglicht es, digitale Zertifikate auszustellen, zu verteilen und zu überprüfen. Besitzer solcher Zertifikate können damit im Web Service Umfeld den Besitz von verschiedensten Claims nachweisen, unter anderem auch ihr Identität. Das Web Service Security Modell unterstützt Security Token Services, die öffentliche asymmetrische Schlüssel verwenden. PKI wird hier im Sinne eines Broadcast eingesetzt, ohne die Annahme einer Hierarchie oder eines speziellen Modells.
- **Kerberos** Das Kerberos Modell ist abhängig von der Kommunikation mit dem Key Distribution Center (KDC) um Vertrauen zwischen Parteien zu vermitteln. Dies wird mittels der Ausgabe von symmetrischen Schlüsseln, die für den jeweiligen Kommunikationspartner verschlüsselt werden und dem gegenseitigen Bekanntmachen der beiden Parteien, gewährleistet. Das Web Service Modell baut auf dem Kerngedanken dieses Modells auf. Auch hier gibt es Security Token Dienste, die Vertrauen vermitteln durch die Ausgabe von Security Token mit verschlüsselten symmetrischen Schlüsseln und verschlüsselten Statements.

Wir sehen, dass die Web Service Security nicht losgelöst von anderen Modellen entwickelt wurde, sondern sich gezielt anderer Protokolle bedient, um darauf aufbauend den speziellen Sicherheitsanforderungen von Web Services gerecht zu werden.

Im folgenden betrachten wir nun WS-Security. Dieses Protokoll bildet den Grundbaustein um Web Services sicherer zu machen und liefert die Grundalgen um später mit weiteren Standards diese Sicherheit zu untermauern.

### 4.3.3 WS-Security

IBM, Microsoft und Verisign trieben die Entwicklung von WS-Security voran. Es ging darum, einen einheitlichen Standard zu schaffen, der mehrere nützliche Protokolle verbindet

und einen praktikablen Sicherheitsansatz bietet.

WS-Security stützt sich komplett auf SOAP als Grundlage ab. Dazu bietet es die Integration von XML-Signature und XML-Encryption. Ein weiterer Aspekt ist die Möglichkeit, Security Tokens auf einfache Weise zu übertragen. Bei der Wahl dieser Security Token ist man völlig uneingeschränkt, da diese Schnittstelle als offener Standard festgelegt ist. Es gibt z.B. Kerberos- und X.509-Zertifikate sowie Username/Password.

Um die Anwendung dieses Standards näher zu erklären, folgt nun ein Beispiel von IBM [4]:

```
(001) <?xml version="1.0" encoding="utf-8"?>
(002) <S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
            xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
            xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
(003)   <S:Header>
(004)      <m:path xmlns:m="http://schemas.xmlsoap.org/rp/">
(005)         <m:action>http://fabrikam123.com/getQuote</m:action>
(006)         <m:to>http://fabrikam123.com/stocks</m:to>
(007)         <m:from>mailto:johnsmith@fabrikam123.com</m:from>
(008)         <m:id>uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</m:id>
(009)      </m:path>
(010)      <wsse:Security>
(011)         <wsse:BinarySecurityToken
                      ValueType="wsse:X509v3"
                      Id="X509Token"
                      EncodingType="wsse:Base64Binary">
(012)         MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
(013)         </wsse:BinarySecurityToken>
(014)         <xenc:EncryptedKey>
(015)             <xenc:EncryptionMethod Algorithm=
                        "http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
(016)             <ds:KeyInfo>
(017)               <ds:KeyName>CN=Hiroshi Maruyama, C=JP</ds:KeyName>
(018)             </ds:KeyInfo>
(019)             <xenc:CipherData>
(020)                <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(021)                </xenc:CipherValue>
(022)             </xenc:CipherData>
(023)             <xenc:ReferenceList>
(024)                 <xenc:DataReference URI="#enc1"/>
(025)             </xenc:ReferenceList>
(026)         </xenc:EncryptedKey>
(027)         <ds:Signature>
(028)            <ds:SignedInfo>
(029)               <ds:CanonicalizationMethod
```

```
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(030)             <ds:SignatureMethod
             Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
(031)             <ds:Reference>
(032)                <ds:Transforms>
(033)                   <ds:Transform
                           Algorithm="http://...#RoutingTransform"/>
(034)                   <ds:Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(035)                </ds:Transforms>
(036)                <ds:DigestMethod
                 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(037)                <ds:DigestValue>LyLsF094hPi4wPU...
(038)                 </ds:DigestValue>
(039)             </ds:Reference>
(040)           </ds:SignedInfo>
(041)           <ds:SignatureValue>
(042)                    Hp1ZkmFZ/2kQLXDJbchm5gK...
(043)           </ds:SignatureValue>
(044)           <ds:KeyInfo>
(045)               <wsse:SecurityTokenReference>
(046)                   <wsse:Reference URI="#X509Token"/>
(047)               </wsse:SecurityTokenReference>
(048)           </ds:KeyInfo>
(049)         </ds:Signature>
(050)      </wsse:Security>
(051)   </S:Header>
(052)   <S:Body>
(053)      <xenc:EncryptedData
                Type="http://www.w3.org/2001/04/xmlenc#Element"
                Id="enc1">
(054)         <xenc:EncryptionMethod
           Algorithm="http://www.w3.org/2001/04/xmlenc#3des-cbc"/>
(055)         <xenc:CipherData>
(056)            <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(057)            </xenc:CipherValue>
(058)         </xenc:CipherData>
(059)      </xenc:EncryptedData>
(060)   </S:Body>
(061) </S:Envelope>
```

Um dieses Beispiel besser verstehen zu können, folgen nun einige Erläuterungen:

- Zeile 1 - 2: wir erkennen, dass wir es mit einem XML-Dokument zu tun haben, in das eine SOAP Nachricht eingebettet ist.
- Zeile 3: hier beginnt der SOAP-Header.

- Zeile 10 - 50: hier befindet sich das Security Element. Dadurch wird signalisiert, dass hier WS-Security angewendet wird.
- Zeile 11 - 13: man sieht den ersten Anwendungsfall: es wird ein Security Token transportiert. Es ist ein X.509-Zertifikat in Version 3, welches in Base64Binary codiert ist.
- Zeile 14 - 26: die Definition für den Schlüssel, mit dem später der Body verschlüsselt wird. Dieser Schlüssel ist ein symmetrischer Schlüssel und wird daher verschlüsselt transportiert.
- Zeile 27 - 49: hier wird die digitale Sigantur spezifiziert.
- Zeile 52 - 60: hier befindet sich der Body der Nachricht.
- Zeile 53 - 59: an dieser Stelle befinden sich die mittels XML Encryption verschlüsselten Daten.

### 4.3.4 WS-Security Erweiterungen

Im letzten Abschnitt wurde der WS-Security Standard erläutert. Dieser wird in dieser Form schon genutzt und hat mittlerweile auch eine gewisse Akzeptanz erreicht. Daher gibt es nun Überlegungen bzw. Anstrengungen Sicherheitsstandards zu entwerfen, die WS-Security als Grundlage verwenden und neue Sicherheitsfeatures bieten können. Diese neuen Standards sollen in diesem Abschnitt kurz vorgestellt werden. Die neuen Erweiterungen sind noch im Prozeß der Standisierung, wurden aber schon implementiert und funktionieren. Die Abbildung liefert einen Überblick über die verschiedenen Entwick-

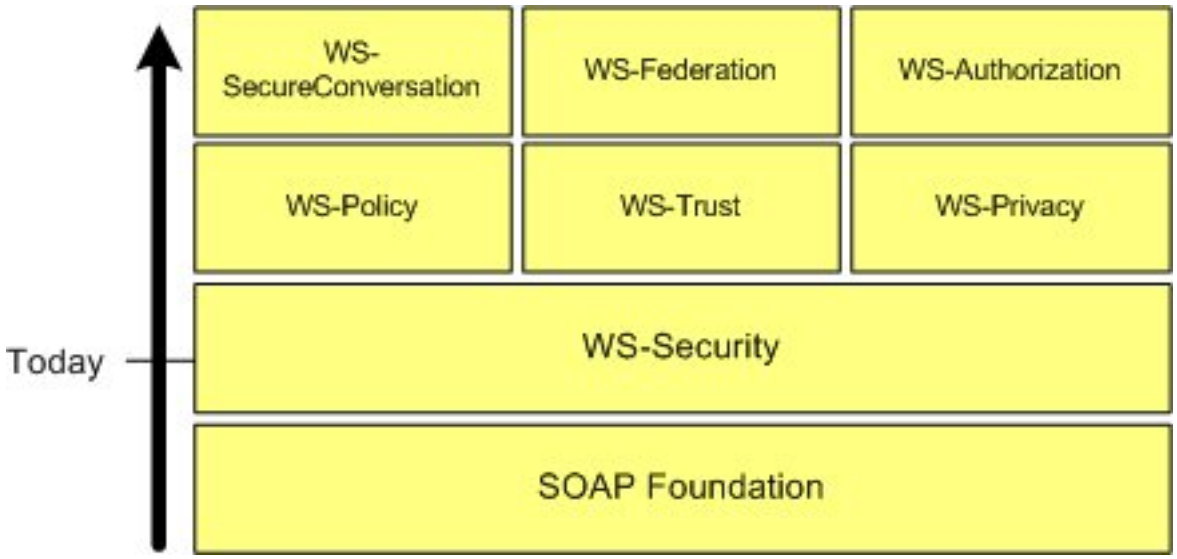


Abbildung 4.2: der WS-Security Stack von [2]

lungen. Man sieht, dass SOAP die Grundlage ist, auf der WS-Security aufbaut. Dann kommt eine neue Schicht mit WS-Policy, WS-Trust und WS-Privacy. Erst wenn diese

zufriedenstellend funktionieren ist es möglich die drei Protokolle der höchste Ebene WS-SecureConversation, WS-Trust und WS-Authorization zu nutzen.

### WS-Policy

Dieser Standard soll eine Möglichkeit für Zwischen- und Endpunkte bieten, eigene Sicherheitsregeln zu definieren und durchzusetzen. Dazu kann man beispielsweise definieren welche Token der Knoten verarbeiten kann, welche Verschlüsselungsalgorithmen verwendet werden oder wie sich authentifiziert werden muss. Um diese Regeln weiterzugeben, stützt sich WS-Policy auf Nachrichtenebene auf WS-Security ab.

```
<wsp:Policy>
  <wsp:ExactlyOne>
  <wsse:SecurityToken>
  <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
  </wsse:SecurityToken>
  <wsse:SecurityToken>
  <wsse:TokenType>wsse:X509v3</wsse:TokenType>
  </wsse:SecurityToken>
  </wsp:ExactlyOne>
</wsp:Policy>
```

An diesem Beispiel kann man sehen, wie einfach sich eine Richtlinie mit WS-Policy definieren lässt. Hier wird demonstriert, wie man eine Regeln festlegen kann, die entweder ein Kerberos Token oder ein X.509 Token annimmt.

### WS-Trust

Bedingt durch den fehlenden persönlichen Kontakt bei Geschäftsbeziehungen im Internet werden Methoden benötigt, welche die Etablierung von Trust (Vertrauen) zwischen Geschäftspartnern ermöglichen. Ziel von WS-Trust ist es, individuellen Sicherheitsbedürfnissen der Dienstnutzer und Dienstanbieter gerecht zu werden. Das Risiko, dass Agenten Dienste nutzen, zu denen Ihnen die Berechtigung fehlt, soll minimiert werden.

WS-Trust beschreibt ein Modell, mit dessen Hilfe sowohl direkte als auch indirekte Trust-Beziehungen aufgebaut werden können. Der Austausch von Security Token findet zwischen drei Parteien statt: dem Dienstnutzer, dem Web Service sowie dem Security Token Service. Auch diese Security Token Service kann ein Web Service sein. Nachdem ein Nutzer bzw. Web Service dem Security Token Service ein von ihm benötigtes Security Token vorgelegt hat, wird der anfragenden Partei wiederum ein Security Token ausgestellt. Eine direkte Trust-Beziehung kann zum Beispiel mittels Benutzername und Passwort unter Verwendung von TLS etabliert werden. Es wird vorausgesetzt, dass beide Parteien ein geteiltes Geheimnis besitzen (z.B. das Userpasswort). Der Benutzer öffnet eine sichere Verbindung über TLS zum Web Service. Seine Anfrage ist ein Security Token beigefügt, welches seinen

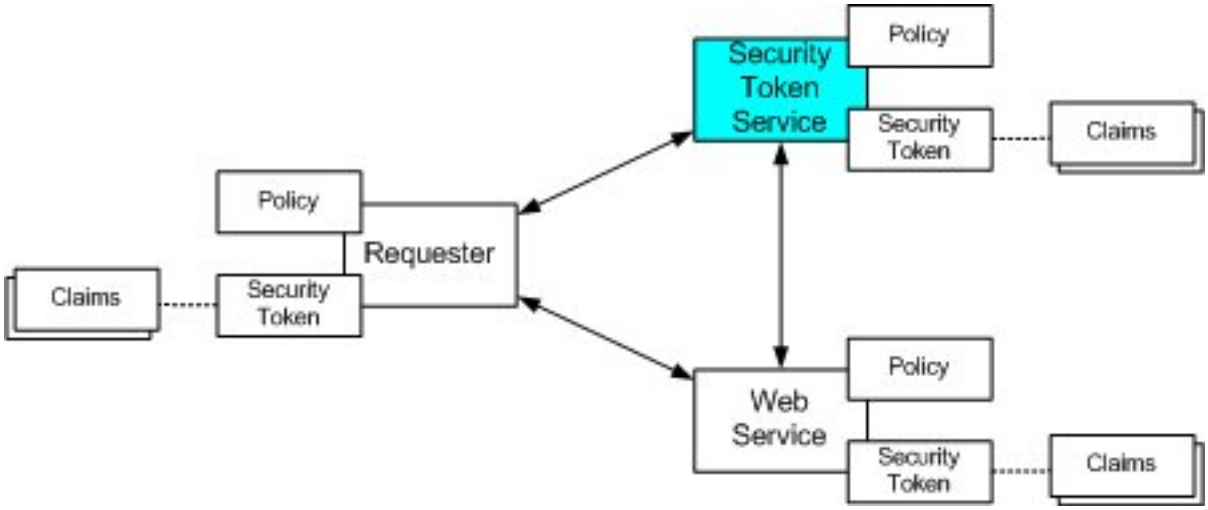


Abbildung 4.3: das Web Service Trust Modell von [2]

Benutzernamen und Passwort enthält. Nachdem der Web Service des Token authentisiert hat, verarbeitet dieser die Anfrage und gibt das Ergebnis zurück.

Eine indirekte Vertrauensbeziehung wird durch die Authentisierung durch eine vertrauenswürdige Instanz realisiert werden. Diese Form des Aufbaus spielt immer dann eine Rolle, wenn ein Web Service ein Security Token eines bestimmten Typs vorraussetzt. Dazu nimmt der Dienstnutzer zunächst Kontakt zu einem Security Token Service auf, um ein Security Token zu erhalten, welches ihm die vom Web Service benötigten Informationen attestiert. Dieses sendet er, zusammen mit seiner Anfrage, an den Web Service, dessen Dienst er auf diese Weise nutzen kann.

WS-Trust ist für die Verwaltung der Vertrauensbeziehungen zuständig. Die Übertragung der Token findet mit Hilfe von WS-Security statt.

### WS-Privacy

WS-Privacy hilft sowohl Dienstanbieter als auch Dienstnutzer, seine Datenschutzbedürfnisse bzw. -angebote zu beschreiben und somit vergleichen zu können. Dies hilft vor allem dem Dienstnutzer sicher zu sein, dass seine aufgestellten Regeln nicht verletzt werden.

So kann ein Agent zum Beispiel seine Privacy Präferenzen und Wünsche beschreiben und mit seiner Anfrage an einen Web Service versenden. Der Web Service vergleicht diese Regeln mit seinen Privacy Practice Rules, die seine Richtlinien und Vorgehensweise bezüglich Datenschutz beschreiben, und entscheidet dann, ob der Web Service ausgeführt werden kann, ohne Datenschutzbedürfnisse zu verletzen.

In der technischen Umsetzung werden diese Privacy Richtlinien in WS-Policy Definitionen eingebettet.

### WS-Authorization

Um Anmeldevorgänge zu erleichtern, wird der WS-Authorization Standard entworfen. Mit seiner Hilfe kann man Anmeldevorgänge spezifizieren und später dann verwalten. Auch hier werden die benötigten Daten via WS-Security ausgetauscht.

### WS-SecureConversation

WS-Security bietet Sicherheit auf Nachrichtenebene. Das bedeutet, dass bei einer längeren Kommunikation jede einzelne Teilnachricht verschlüsselt, signiert und gegebenenfalls mit einem Security Token versehen wird. Diesen Overhead will WS-SecureConversation bekämpfen, indem es Mechanismen zur Sicherung mehrerer Nachrichten zwischen zwei Parteien bietet. Es wird ein Sicherheitskontext geschaffen, um eine sichere Verbindung nutzen zu können. Dies passiert mittels WS-Trust, um die Vertrauensbeziehung herzustellen und dieses Protokoll nutzt wiederum WS-Security. WS-SecureConversation definiert hierbei, wie Services (z.B. mit welchem Key) während der Dauer einer Session Daten sicher untereinander austauschen können.

### WS-Federation

In diesem Abschnitt wird gezeigt, wie sich mit Hilfe von WS-Federation Trust Verhältnisse zwischen mehreren Parteien anwenden lassen. Zwei dieser Möglichkeiten sollen hier anhand eines Beispiels von Microsoft [2] gezeigt werden.

Der erste Ansatz geht davon aus, dass der Währungs-WS ausschließlich Security Token akzeptiert, die von Business456 herausgegeben wurden. Über die Policy des Web Service erfährt Alice, wo sie das benötigte Security Token erhalten kann. Alice legt dem Business456 Security Token Service ihr Adventure456 Security Token (inklusive Eigentumsnachweis) vor und erhält von ihm ein Business456 Security Token. Mit diesem kann sie nun eine Anfrage an den Währungs-WS stellen. Abb. 1.4 oberer Teil veranschaulicht dieses Szenario.

Im zweiten Ansatz wird das Bündnis mittels einer Vertrauenskette realisiert. Der Währungs-WS ist in diesem Fall so eingerichtet, dass er Anfragen mit allen möglichen Security Token akzeptiert, diese jedoch erst dann verarbeitet, wenn er im Tausch für das erhaltene Token ein Business456 Security Token erhalten hat. Dabei geht der Währungs-WS wie folgt vor: Die Anfrage des Dienstnutzers, sowie dessen Security Token, wird an den Business456 Security Token Service weitergeleitet, der das Token auswertet. Falls er es als gültig akzeptiert wird die Anfrage gebilligt und der Security Token Service übermittelt möglicherweise auch ein Business456 Security Token, welches Alice später ggf. wiederverwenden kann. Zu beachten ist in beiden Ansätzen, dass der Security Token Service von Business456 so eingerichtet ist, dass er den Identitätsnachweis, der von Adventure456 herausgegeben wurde, akzeptiert. Abb. 1.4 unterer Teil veranschaulicht dieses Szenario.

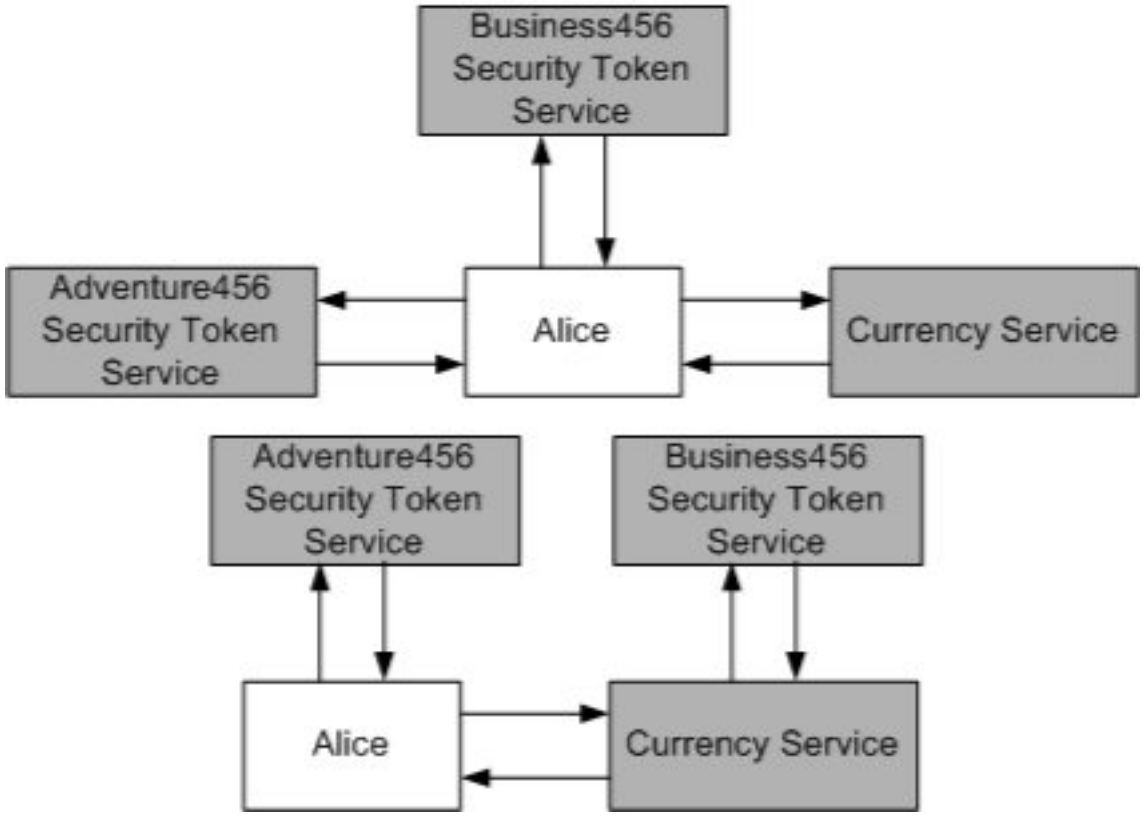


Abbildung 4.4: zwei Beispiele zu WS-Federation von [2]

### 4.3.5 Sicherung der Interorperabilität

Um die Interorperabilität von Web Services zu sicher gibt es die Web Service Interoperability Organization, kurz WS-I. Es ein offener Verbund von mehr als 130 Mitgliedsorganisationen, die versuchen Standards in Web Service zu integrieren und damit die Entwicklung von Web Service auf einem strukturierten Weg fortzuführen.

Dadurch ergeben sich drei große Ziele: die Interoperabilität erreichen, die Entwicklung von Web Services beschleunigen und die Übernahme von Web Services in bestehende Systeme voranzutreiben. Dazu werden Profiles herausgegeben, die Richtlinien zur gemeinsamen Nutzung verschiedener Spezifikationen enthalten. Danaben gibt es Beispielapplikationen, Test Werkzeuge und weiteres Material, die helfen schneller und verläßlicher Web Services einzurichten.

WS-I sieht sich selbst als eine Art Zwischenschicht. Sie befinden sich zwischen dem W3C oder OASIS, die Standards und Spezifikationen festlegen und dem Endanwender, wie zum Beispiel die Wirtschaft. Sie möchten dem Endanwender „gebündeltes“ Wissen liefern und durch die Praxiserfahrungen den Standardisierungsgremien fundiertes Feedback geben zu können.

# Literaturverzeichnis


[1] Eberhart, Andreas: *Web Services*
2003 Carl Hanser Verlag; ISBN 3-446-22530-7

[2] WS-Security Roadmap von Microsoft
*msdn.microsoft.com/library/*

[3] Web Service Activity Internetseite des W3C
*http://www.w3.org/2002/ws/*

[4] Web Service Security Spezifikation von IBM
*http://www-106.ibm.com/developerworks/webservices/library/ws-secure/*

[5] Schirru, Rafael: Seminarabeit zum Thema Trust, Reputation, Privacy
*http://wwwdvs.informatik.uni-kl.de/courses/seminar/SS2004/rschirrua.pdf*

[6] Begriffsdefinitionen
von *www.wikipedia.org*

# Kapitel 5

# Webservices - Anwendung, Verbreitung, Nutzung

*Kai Freytag*

*In diesem Kapitel werden die beiden wichtigsten Plattformen zur Entwicklung von Webservices vorgestellt und erläutert. Es wird aufgezeigt, wie die Architekturen der beiden Plattformen aufgebaut sind und wie Webservices auf Basis dieser Plattformen entwickelt werden. Im Besonderen richtet sich der Augenmerk darauf, wie die entsprechenden Plattformen Standards wie WS-Security unterstützen.*
*Daraufhin werden exemplarisch, anhand von ausgewählten Webservice-Werkzeugen, die Möglichkeiten dieser Werkzeuge untesucht, sichere Webservices auf Grundlage unterschiedlicher offener Standards zu erstellen. Sie sollen im Idealfall Entwickler in die Lage versetzen, Webservices nicht nur zu entwickeln, sondern auch entsprechend bestimmter Standards, im Besonderen WS-Security , abzusichern. Es wird bei der Untersuchung der Werkzeuge anhand der Klassifizierung der Entwicklungsplattformen in J2EE-Toolkits und .NET-Toolkits unterschieden.*
*Schließlich werden einige Anwendungsfälle der WS-*-Spezifikation vorgestellt und erläutert. Anschließend wird die praktische Umsetzung dieser Spezifikationen, anhand von Projekten, die tatsächlich in der Industrie so existieren, betrachtet. Abschließend werden die Erfahrungen und Schlußfolgerungen aus diesen praktischen Anwendungen zusammengefasst und präsentiert.*

## Inhaltsverzeichnis

## 5.1 Entwicklungsplattformen für Webservices

### 5.1.1 Die J2EE-Plattform

Java 2.0 Enterprise Edition (J2EE) ist ein offener Standard auf Basis der Programmiersprache Java von Sun Microsystems. Im Markt der *Application Server* hat sich J2EE als herstellerübergreifender Implementierungsstandard durchgesetzt und J2EE-konforme *Application Server* werden heute von einer ganzen Reihe von Herstellern angeboten.
Ein Webservice ist eine Serveranwendung, die als Container implementiert ist. Dieser Container enthält diejenigen Methoden, die für einen Client per SOAP zugänglich sind. Es kann dazu entweder ein Web Container wie Apache Tomcat, oder ein *Application Server* verwendet werden. Die grundlegende Technologie, die hinter beiden Varianten steckt, basiert auf der *Java API for XML-based RPC* (JAX-RPC).
Der entscheidende Faktor für Webservices ist deren Interoperabilität. Genau dies wird durch JAX-RPC und die darin enthaltene Unterstützung von SOAP und WSDL gewährleistet. [1]

#### Web Container mit Apache AXIS

Apache AXIS (Apache eXtensible Interaction System) ist eine SOAP Engine, also ein Rahmenwerk zur Konstruktion von darauf basierenden Webservices und Client-Anwendungen. Es existiert eine Implementierung in Java und C++.
Apache AXIS ist eine Neuentwicklung und Nachfolger von Apache SOAP, das auf dem IBM-Framework SOAP4J basierte. AXIS wird häufig als Java-Servlet innerhalb eines Servlet-Containers (beispielsweise Apache Tomcat) betrieben. Die eigentliche Funktionalität der Webservices ist in Java-Klassen gekapselt und wird dem Webservice-Konsumenten über das AXIS-Servlet zur Vefügung gestellt.
Mit den in AXIS integrierten Tools JAVA2WSDL und WSDL2JAVA wird der Entwickler unterstützt, eine robuste Schnittstelle in Java zu erzeugen, ohne sich direkt mit SOAP befassen zu müssen.
Für einfache Anwendungen bietet Apache AXIS an, dass Webservices direkt als Java-Klassen bzw. JWS (Java Webservices) bereitgestellt werden können. Aus diesen Klassen generiert AXIS automatisch einen verwendbaren Webservice inklusive WSDL-Beschreibung, die über das Netz ausgelesen werden kann, um hierfür einen entsprechenden Client zu entwickeln. Für komplexere Webservices sollte ein sog. *Webservices Deployment Descriptor* (WSDD) verwendet werden. Er enthält Metadaten über den Webservice, die Apache AXIS verwendet, um den internen Ablauf eines Webservice-Aufrufs zu steuern.[2]

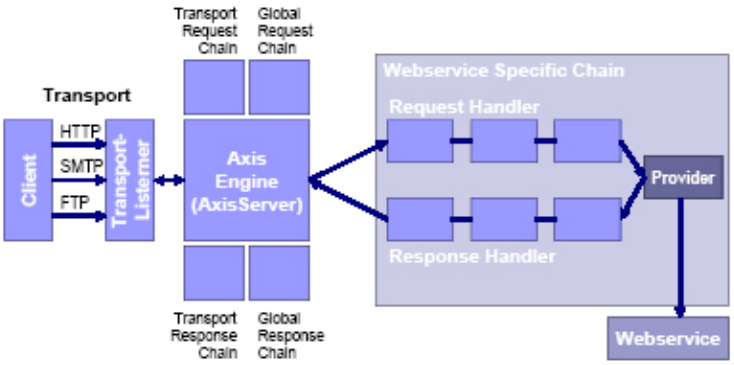


Abbildung 5.1: Architektur von Apache AXIS

Ein *Application Server* muss, wenn er SOAP-fähig sein soll, in der Lage sein mit unterschiedlichen Transportprotokollen, wie in Abbildung 5.1 angedeutet, zu kommunizieren. Die Transport-Komponente ist verantwortlich für das Versenden und den Empfang von SOAP-Anfragen bzw. SOAP-Antworten. Sie dient aber auch als Schnittstelle für die physikalischen Netzwerkkomponenten.
Die Verarbeitung von Anfragen erfolgt über sogenannte Handler-Ketten, die sich aus einzelnen Handlern zusammensetzen. Client und Server bestehen immer aus einer AXIS Engine, welche die Handler-Ketten Service, Transport und Global umfaßt. Die Verkettung der Handler ist flexibel und kann über Konfigurationsdateien angepaßt werden. Typische Aufgaben sind Logging oder die Unterstützung von Sicherheitsmechanismen wie WS-Security. Der Entwickler kann fertige Handler für Logging, Authentifizierung oder das SOAP Monitortool einbinden. Wird Funktionalität benötigt, die über die mitgelieferten Handler hinausgeht, ist AXIS offen für eigene Erweiterungen, die sich recht einfach realisieren lassen. Ein Handler wird mit einer Klasse erstellt, die das Handler-Interface implementiert. Die abstrakte Klasse BasicHandler erleichtert die Erstellung von Handlern über Vererbung. Die Callback Methoden *invoke()* und *onFault()* können mit eigener Funktionalität überschrieben werden. Schließlich erfolgt die Serialisierung, die für das Datentyp-*Mapping* von XML auf Java und umgekehrt zuständig ist (JAX-RPC konform).
Nachdem die Anfrage nun aufbereitet worden ist, wird sie durch den *Service Dispatcher* (Provider) an den vorgesehenen Webservice weitergeleitet. Der Dienst wird ausgeführt und schließlich wird das Ergebnis als SOAP-Nachricht serialisiert und über die Transport-Komponente an den entsprechenden Client zurückgesendet. Die Verarbeitung auf dem Client bzw. Client-Proxy verläuft analog zu dem beschriebenen Ablauf auf der Server-Seite.[3]

### Application Server

Ein *Application Server* ist ein Server in einem Computernetzwerk, auf dem eine spezielle Softwareanwendung läuft. Häufig meint man dabei Software-Applikationen mit einer drei- oder mehrschichtigen Architektur. Ziel ist es, die drei Aufgaben Präsentation, Geschäftslogik und Datenhaltung voneinander zu trennen.[4]
Zu den wichtigsten kostenpflichtigen J2EE Application Servern zählen: *WebLogic* von Bea, *WebSphere* von IBM und *Oracle9i AS* von Oracle. Zur Zeit werden *WebLogic* und *WebSphere* in Unternehmen am häufigsten eingesetzt. Bezüglich Stabilität und Performance

gibt es keine großen Unterschiede mehr.
Weitere *Application Server* wie *Sun ONE AS* oder *Borland AS* bemühen sich um höhere Marktanteile, jedoch bisher ohne große Chancen. Inzwischen gibt es einige freie *J2EE Application Server*, die es bezüglich Zuverlässigkeit und Performance mit den kostenpflichtigen durchaus aufnehmen können. Zu den wichtigsten zählen *JBoss* und *Jonas.*[5] *Application Server* bieten im Allgemeinen mehrere Möglichkeiten, um Webservices abzusichern. Die beiden wichtigsten Methoden sind einmal der Einsatz von *Web-Service-Security-Gateways*, die SOAP-Nachrichten rekonstruieren und auf bösartigen Inhalt hin prüfen. Zum Anderem können *Application Server*, ebenso wie die bereits beschriebenen Web Container, Standards wie die WS-*-Spezifikation unterstützen.

### 5.1.2 Die .NET-Plattform

Die .NET-Plattform ist ein Bündel von Softwaretechnologien des Herstellers Microsoft, das als Gegenpol zu Sun Microsystems J2EE-Standard eingeführt wurde. Es besteht neben einer virtuellen Laufzeitumgebung aus einem Framework von Klassenbibliotheken (API) und Diensten, die als Basis für Eigenentwicklungen dienen.
Die .NET-Plattform stellt mit der *Common Language Infrastructure* (CLI) eine Basis zur Ausführung von Programmen, die mit unterschiedlichen Programmiersprachen erstellt wurden, her. Dies wird durch die Verwendung einer objektorientierten, virtuellen Maschine und die *Framework Class Library* (FCL), einer gemeinsamen Klassenbibliothek, erreicht.
Neben den von Microsoft für die .NET-Plattform angepassten Sprachen wie C#, Visual Basic.NET und Managed C++ werden weitere .NET-Sprachen von Drittanbietern zur Verfügung gestellt (zum Beispiel Delphi und Borland).
Die integrierte Entwicklungsumgebung (IDE) Visual Studio.NET, die von Microsoft für die .NET-Plattform optimiert worden ist, ist ein integraler Bestandteil des .NET-Frameworks. Diese IDE unterstützt alle Sprache, die die .Net-Sprachenspezifikation erfüllen und somit ist die Entwicklung eigener IDEs für jede Sprache unnötig.
Microsoft bietet das .NET-Framework neben der kostenpflichtigen IDE Visual Studio.NET auch in zwei weiteren Formen an: Zum Einen als reine Laufzeitumgebung samt benötigter Klassenbibliotheken und zum Anderen als kostenloses SDK für Entwickler.[6]

#### Das .NET-Framework

Die Framework Class Library (FCL) umfasst einige tausend Klassen, die in sogenannte Namensräume (Namespaces) unterteilt sind. Die Klassen erfüllen Aufgaben, wie das Formatieren von Text oder das Generieren von Code. Die Unterteilung in Namensräume dient dazu die große Menge an Information übersichtlicher zu gestalten.

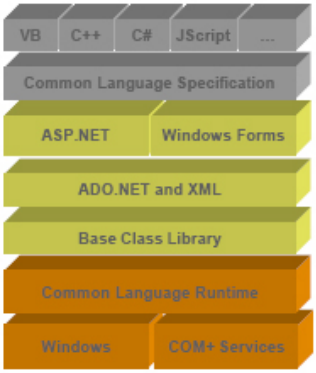


Abbildung 5.2: Das .NET-Framework

Die Basisklassen (Base Framework Classes) bieten Standardfunktionalität, wie zum Beispiel Eingabe/Ausgabe, Text- und Sicherheitsverwaltung. Die Daten- und XML-Klassen bieten Unterstützung der Verwaltung von dauerhafter Datenbestände an und umfassen auch SQL-Klassen (ADO.NET), sowie die Bearbeitung und das Übersetzen von XML-Dateien. Die *XML Webservices* und *Web Forms*-Klassen (ASP.NET) unterstützen die Entwicklung von Web-Anwendungen und Webservices. Darauf wird im nächsten Abschnitt genauer eingegangen. Schließlich unterstützt die *Windows Forms*-Klassen bei der Erstellung von Windows-basierenden Client-Anwendungen.[7]

### Webservices mit ASP.NET erstellen

Die Implementierung von Webservices mit der .NET-Plattform kann mit allen von .NET unterstützten Programmiersprachen geschehen. Es wird dabei SOAP durch die Plattform genutzt, ohne das der Entwickler sich näher damit beschäftigen muss. Die gesamte Webservice-Infrastruktur ist bereits durch .NET implementiert, so dass der Entwickler lediglich die Funktionalität des Webservice zu programmieren braucht.
Die Sicherheit für ASP.NET Webanwendungen beginnt mit dem *Internet Information Service* (IIS), einer Erweiterung des Windows Server. Als Server auf Windows-Basis ist IIS vollständig in die Windows-Sicherheit integriert.
Eine Webanwendung wird in der Regel mit ASP.NET als Web-Engine ausgeführt, das über eigene Sicherheitseinrichtungen verfügt. Diese kommen zum Einsatz, wenn die Anwendung Zugriff auf Ressourcen verlangt. Wenn Sie in der Webanwendung beispielsweise eine Datei lesen oder schreiben möchten, ist der ASP.NET-Sicherheitskontext dafür verantwortlich, ob die Anforderung erfolgreich verläuft. Nicht alle Benutzer haben jedoch die erforderliche Authentifizierung zum Lesen von Dateien auf einem Windows-Server. Dies ist insbesondere bei Webanwendungen der Fall, die öffentlich im Internet verfügbar sind. IIS und ASP.NET bietet daher mehrere Mechanismen zum Herstellen einer Authentifizierung, die WS-Security aber nicht unterstützen. Dafür ist, wie später genauer erläutert wird, eine Erweiterung des .NET-Frameworks nötig, die Microsoft mit den *Web Services Enhancements* (WSE) liefert.[8]
Um Webservices schließlich auszuführen, muss ASP.NET am Server installiert sein und zur Erstellung der Anwendung benötigt man wenigstens das vollständige .NET SDK. Der erzeuge Code des Webservice wird dann als *asmx*-Datei im entsprechenden Verzeichnis

im Server abgelegt. Um schließlich zu zeigen, dass der Webservices die gewünschte Funktionalität bietet oder den um den Webservice lediglich zu nutzen, generiert ASP.NET für jeden Webservice Standard HTML-Seiten, die es erlauben, die Definition des Services einzusehen (WSDL), als auch die freigegebenen Methoden aufzurufen.[9]

### 5.1.3 Weitere Entwicklungsmöglichkeiten

Es gibt neben den J2EE- und .NET-Technologien noch einen weiteren Ansatz Webservices zu erstellen: PEAR („PHP Extension and Application Repository“) PEAR ist eine Sammlung bzw. Bibliothek von Modulen und Erweiterungen für die Skriptsprache PHP. Die Programmmodule und Erweiterungen verkürzen den Entwicklungsprozess von PHP-Anwendungen erheblich, da sie dem Programmierer viele Standard-Aufgaben, wie die SOAP-Kommunikation, abnehmen. Hierdurch wird auch ein höheres Maß an Sicherheit bei der Anwendungsentwicklung erzeugt, da der Entwickler auf qualitätsgesicherte Standardkomponenten zurückgreift.
Alle PEAR-Projekte stehen als Open Source zur Verfügung und können mit dem PEAR Installer sehr einfach auf dem eigenen Webserver installiert werden.[10]

## 5.2 Werkzeuge zum Erstellen sicherer Webservices

Im vorherigen Abschnitt wurden die Plattformen vorgestellt, die es ermöglichen Webservices zu entwickeln und zur Verfügung zu stellen. Nun wird untersucht, welche Werkzeugunterstützung Entwicklern für die jeweilige Plattform zur Verfügung steht, um Webservices entsprechend existierender Sicherheitsstandards, wie *WS-Security* abzusichern.

### 5.2.1 J2EE-Toolkits

Im Bereich der Java-basierter Webservice-Entwicklung wird das *Java Web Service Developement Pack* (JWSDP) von Sun Microsystems und das *Trust Service Integration Kit* (TSIK) von VeriSign exemplarisch näher betrachtet. Es gibt neben diesen Open Source-Werkzeugen noch viele weitere, zum Teil kommerzielle, Toolkits, wie beispielsweise das WSTK von IBM.

#### Sun JWSDP

Das JWSDP ist ein kostenloses SDK zum Entwickeln von Webservices und Java-basierten Anwendungen unter Verwendung der neuesten Java-Technologien. Das Toolkit besteht unter anderem aus der folgenden Sammlung von Java-Bibliotheken (API‘s) :[11]

- *Java API for XML Processing* (JAXP): Einheitliches API, um unter Java auf XML zuzugreifen. Umfasst DOM, SAX und XSLT und kann verschiedene XML-Parser einbinden.

- *Java Architecture for XML Binding* (JAXB) : Definiert einen Mechanismus zum Schreiben von Java-Objekten als XML-Dokument (Marshalling) und zum Erzeugen von Java-Objekten aus XML-Dokumenten (Unmarshalling).
- *Java API for XML-Based Remote Procedure Call* (JAX-RPC) : Definiert einen Mechanismus zum Austausch synchroner XML-Messages als *Remote Procedure Calls*, bei denen ähnlich wie bei Funktionsaufrufen auf das Ergebnis gewartet wird.
- *SOAP with Attachments API for Java* (SAAJ) : Stellt einen Standard-Mechanismus zu Verfügung, mit dem XML-Dokumente über das Internet von der Java-Plattform verschickt werden kann.
- *Java API for XML Registries* (JAXR) : Definiert einen Mechanismus zum Veröffentlichen verfügbarer Dienste in einer externen *Registry* und zur Suche von Diensten in einer solchen *Registry* (UDDI oder ebXML *Registry*).[12]

Zusätzlich gibt es noch weitere Komponenten in diesem Toolkit, so dass es sich tatsächlich um eine vollständige Entwicklungsumgebung für Webservices und für dynamische Web-Inhalte handelt. Ein wichtiger Bestandteil des JWSDP ist die Integration der *JSP Standard Tag Library* (JSTL) aus dem Jakarta-Projekt. JSTL trennt die Programmlogik aus einer JSP-Seite heraus. Web Designer können somit ohne Java Code anspruchsvolle Seiten gestalten. Weiterhin gehört ein *Ant Build Tool* zum Toolkit. Dadurch werden die Prozesse, die zur Herstellung und Wartung einer Anwendung beitragen automatisiert, indem sie in einem zentralen XML-Script abgelegt werden.
Schließlich stellt das JWSDP einen *Java WSDP Registry Server* zur Verfügung. Er stellt eine UDDI-kompatible *Registry* für Webservices in einem privaten Umfeld bereit, das als Testumgebung (Test-*Registry*) für die Webservice-Entwicklung gedacht ist. Das Herzstück des JWSDP ist jedoch die Tomcat Servlet-Engine. Die verwendete Version unterstützt die neuesten Standards, wie JSP 1.2 und Servlet 1.3 API. Zu beachten ist allerdings, dass es sich nicht um einen vollständigen Tomcat handelt, zum Beispiel sind die *Web-Server-Connectors* für eine Verbindung zum Apache Web-Server nicht enthalten.[13]
Die Pakete *XML Digital Signature* und *XML Digital Encryption*, die die Grundlage für WS-Security bilden, sind ebenfalls Bestandteil des JWSDP ab Version 1.5. Somit wird die Unterstützung von WS-Security sichergestellt. [14]

### VeriSign TSIK

Das *Trust Service Integration Kit* (TSIK) wird von VeriSign in Form von Open Source Java APIs zur Verfügung gestellt, die die Entwicklung interoperabeler, vertrauenswürdiger Anwendungen vereinfacht. Entwickler können mithilfe diese Toolkits sichere Webservices schnell integrieren, die unter anderem dem WS-Security-Standard genügen. Features wie XML-Verarbeitung, XML-Authentifizierung, XML-Autorisierung sowie eine Gültigkeitsprüfung von Zertifikaten in Echtzeit sind hierbei mit eingeschlossen.

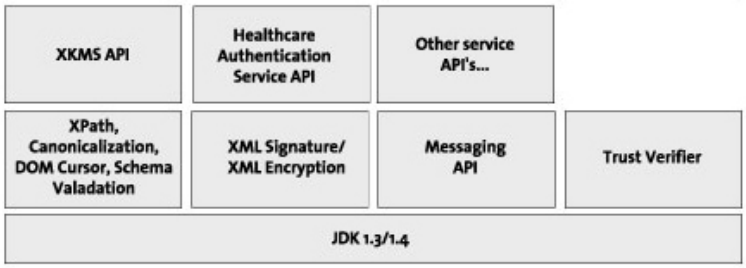


Abbildung 5.3: Trust Services Integration Kit

TSIK ist also ein Java-basiertes Entwicklungswerkzeug, mit dem Sicherheitseigenschaften, wie XML Signature, XML Encryption und XML Key Management Services (XKMS), in Webservices eingebunden werden können. Das TSIK besteht aus drei Basiskomponenten:

1. Das *Messaging Framework*: Zuständig für die Generierung von Signaturen und Kodierungsschlüssel um Authentifizierung, Datenintegrität und Vertraulichkeit sicherzustellen. Das Framework wird erweitert durch *Trust Assertions*, um Autorisierung für die Zugriffsverwaltung zu ermöglichen.

2. Der *Trust Layer*: Stellt APIs für sichere XML-Nachrichten zur Verfügung, die eine *Public Key Infrastructure* (PKI) nutzen. Weiterhin beinhaltet diese Komponente die Implementierungen der W3C-Spezifikationen XML Signature und XML Encryption. Zusätzlich enthalten die APIs eine Schnittstelle, genannt *Trust Verifier*, mit der Entwicklern die Möglichkeit gegeben wird, *Trust Policies* für ihre Anwendungen durchzusetzen.

3. Die XML-APIs: Beinhaltet *low-level* APIs, um direkt XML zu manipulieren, Datentypen zu erstellen, durch Dokumentstrukturen zu traversieren und XML Formate zu validieren.[15]

Durch TSIK können Entwickler Vertrauenseigenschaften schnell und einfach implementieren, indem eine kryptographische Unterstützung für digitale XML-Signaturen und XML-Verschlüsselung mithilfe konventioneller XML-Toolkits eingearbeitet wird. Die XKMS-Komponente von TSIK gibt die mit PKI verbundene Komplexität an die Komponenten der Server-Seite weiter. Dadurch können sich Entwickler auf die Anwendungsentwicklung konzentrieren, anstatt sich mit den Problemen der PKI-Einführung auseinander zu setzen.[16]

### 5.2.2 .NET-Toolkits

Nachdem nun genauer beleuchtet wurde, mit welchen Werkzeugen Java-basierte Webservices absichern werden können, werden nun im .NET Bereich die entsprechenden Toolkits vorgestellt. Betrachtet werden hier das *SOAP-Toolkit* im Allgemeinen und die *Web Service Enhancements* (WSE) von Microsoft im Besonderen.

### Microsoft SOAP-Toolkit

Das MS SOAP-Toolkit ist ein SDK für die Verwendung von SOAP-Kommunikation in Projekten mit *Visual Studio 6.0*. Es stellt Werkzeuge zur Verfügung, mit denen die Implementierung und Verwendungen eines Webservice erleichtert wird.
Das SOAP-Toolkit besteht daher aus folgenden Komponenten: Einem Server-seitigem SOAP-Listener, einer *Remote Object Proxy Engine* (ROPE), der SDL und einem SDL-Sourcecode-Generator (SDL Wizard).
Das Kernstück des SOAP-Toolkit ist ROPE. Es schließt alle benötigen Infrastrukturen ein, mit den eine Client-seitige SOAP-Applikation und einem Server-seitigen SOAP-Listener erstellt werden können. Ein SOAP-Listener kann dann SOAP-Nachrichten empfangen, die von SOAP-Client versendet wurden und dies analysieren und verarbeiten. Mit der neuen Unterstützung für SSL, Standard-Authentifizierung, Integrierte Windows-Authentifizierung und Clientzertifikat-Authentifizierung in der November-Version des SOAP-Toolkits können Entwickler den Zugriff auf Webdienste einfach absichern. Es ist Entwicklern außerdem möglich, Client-Anwendungen zu erstellen, die auf gesicherte Webdienste zugreifen. Die größte Schwierigkeit beim Implementieren eines sicheren Webdienstes liegt darin, die Berechtigungen für alle vom Webservice verwendeten Dateien richtig festzulegen. Wenn die Sicherheit auf Systemebene auf der Basis von Windows-Benutzerkonten für die Dienste nicht funktioniert, ist es relativ einfach, die Sicherheit auf Anwendungsebene zu implementieren, indem die Clientanmeldeinformationen im Rumpf der SOAP-Meldungen übergeben werden.[17]

### Microsoft WSE 2.0

Microsoft erweitert mit *Web Services Enhancements* (WSE) die Unterstützung von Webdiensten im .NET Framework, die auf XML, SOAP und WSDL aufbauen. Die auf höherer Ebene verwendeten Protokolle unterstützen beispielsweise nachrichtenbasierte Sicherheit und richtlinienbasierte Verwaltung. Mit WSE wird der Nachrichtenaustausch flexibel genug, um die rein HTTP-basierte Ebene zu verlassen.
Die neue Version 2.0 baut auf ersten Version auf, die grundlegende Unterstützung für Sicherheit, Nachrichtenrouting und binäre Datenübertragung bot. Im Bereich der Sicherheit unterstützte WSE 1.0 beispielsweise die nachrichtenbasierte Authentifizierung, digitale Signaturen und Verschlüsselung. WSE 2.0 bietet nun eine erweiterte Unterstützung zum Integrieren von *Kerberos-Token* in die Windows-Benutzer und -Gruppen. Zusätzlich wird durch die *WS-SecureConversation*-Spezifikation eine effizientere Verarbeitung erreicht, da ein Sicherheitstoken für den Austausch von mehreren Nachrichten zwischen zwei Einheiten verwendet werden kann.[18]

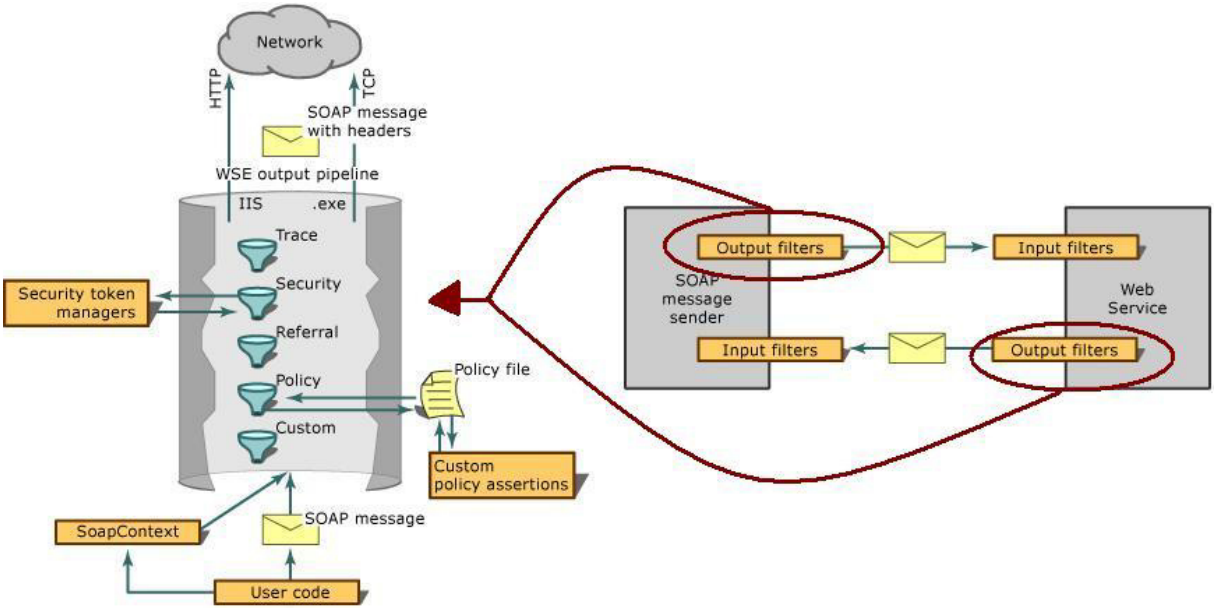


Abbildung 5.4: Prinzip eine WSE Filterkette

Im Grunde genommen ist WSE eine Web-Engine, die fortgeschrittene Webservice-Protokolle auf SOAP-Nachrichten anwendet. Das zieht nach sich, das durch die WSE-Engine XML-Header bei eingehenden SOAP-Nachrichten gelesen und für ausgehende neu geschrieben werden müssen. Ebenfalls kann es notwendig sein, dass der Rumpf einer SOAP-Nachricht gemäß der *WS-Security*-Spezifikation codiert bzw. decodiert werden muss. Diese Funktionalität ist gekapselt in zwei Arten von Filtern, eine für ausgehende und ein für eingehende Nachrichten. Alle Nachrichten, die einen Prozess verlassen (Anfragen von Clients oder Server-Antworten) werden über einen sog. *Outbound-Message-Filter* verarbeitet. Umgekehrt werden alle Nachrichten, die in einem Prozess ankommen (Anfragen an einem Server oder Antworten an einem Client) über einen sog. *Inbound-Message-Filter* verarbeitet. Die Filter bestehen aus Filter-Ketten, über deren Subfilter weitere Feineinstellungen vorgenommen werden können.[19]

## 5.3 WS-Security Anwendungen

Nachdem nun neben den unterschiedlichen Plattformen auch die Werkzeugunterstützung zur Entwicklung und Absicherung von Webservices untersucht wurde, beschäftigt sich dieser Abschnitt mit der Verbreitung und Nutzung dieser Technologien in der Praxis. Zunächst werden einige Anwendungsfälle betrachtet und schließlich wird gezeigt, wie diese Anwendungsfälle in realen Anwendungen umgesetzt worden sind.

### 5.3.1 Anwendungsfälle für WS-Security

In diesem Abschnitt werden verschiedene Anwendungsfälle (Use Cases) der WS-*-Spezifikation beschrieben. Die Komplexität und Umfang der Szenarien nimmt dabei stetig zu. Im letzten Anwendungsfall wird dann ein konkreter Geschäftsvorgang beschrieben.

### Sichere Verarbeitung durch eine Firewall

Wie unten abgebildet, untersucht die Firewall eingehende SOAP-Nachrichten und erlaubt lediglich denjenigen Nachrichten, die von dem autorisierten Anfragesteller stammen, die Firewall zu durchqueren.

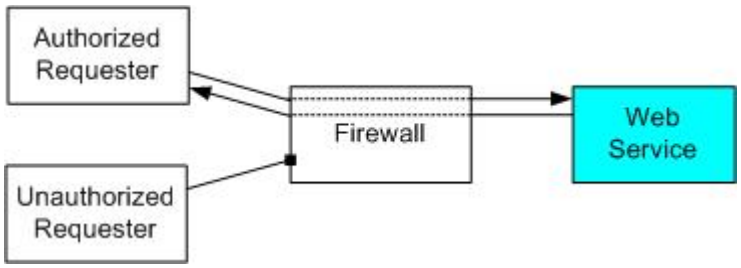


Abbildung 5.5: Sichere Verarbeitung durch eine Firewall

In diesem Szenario sind also zwei Akteure, die Nachrichten zu einem Webservice durch eine Firewall senden. Die Firewall prüft die Nachrichten, um festzustellen, ob der Sender autorisiert ist, mit dem entsprechenden Webservice hinter der Firewall Kontakt aufzunehmen.
Diese Entscheidung der Firewall basiert auf der Untersuchung der *Security Token*, mit denen die SOAP-Nachricht gemäß WS-Security signiert ist. Ist die Signatur des Tokens gültig, der Instanz, die das Token ausgestellt hat, vertraut wird und das Token den Sender autorisiert dem entsprechenden Webservice Nachrichten zu schicken, wird die Nachricht durchgelassen. In jedem anderen Fall wird sie zurückgewiesen.
In einem etwas abgewandelten Szenario, wäre es denkbar, das sich die Firewall selbst wie eine *Security Token*-Zertifizierungsstelle verhält und nur die Nachrichten durchläßt, die ein gültiges Token der Firewall besitzen.

### Durchsetzung von Geschäftsrichtlinien

In vielen Geschäftsprozessen gibt es bestimmte Richtlinien die eingehalten werden müssen, um den Prozess korrekt abzuwickeln. Ein Dienst beispielsweise könnte verlangen, das seine Konsumenten eine bestimmte Bewertung besitzen oder bestimmte *Security Token* benutzen. Mit Webservices ist es möglich das solche Richtlinien automatisch validiert und durchgesetzt werden und somit der gesamten Geschäftsprozess an Komplexität verliert.

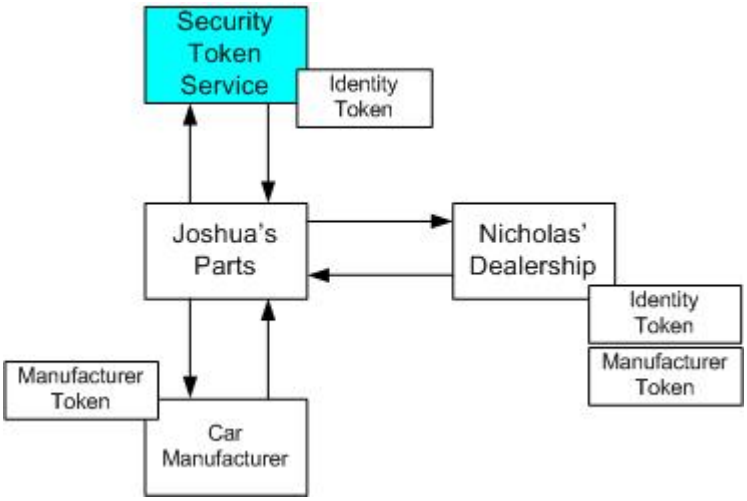


Abbildung 5.6: Durchsetzung von Geschäftsrichtlinien

In diesem Beispiel besitzt ein Autoverkäufer einen Webservice, der genutzt wird, um mit seinen Zulieferern für Ersatzteile zu interagieren. Der Autoverkäufer möchte nur von solchen Zulieferern Autoteile erwerben, die ein Zertifikat von einem bestimmten Autohersteller besitzen. Also wird sich ein solcher Betrieb (Joshua's Parts) zunächst mit einem *Security Token* versorgen, das seine Identität nachweist und schließlich ein Zertifikat vom Autohersteller erhalten (vorausgesetzt seine Ersatzteile stammen von dem Hersteller). Der Zulieferbetrieb kann sich nun mit dem Autoverkäufer in Verbindung setzen und beide *Security Tokens* vorweisen und somit liefern.[20]

### Mehrstufige Webservices

Abschließend sollen nun das Szenario eines mehrstufigen Webservice vorgestellt werden, das in Kombination mit WS-Security und SAML abgesichert wird. Die Abbildung 5.7 zeigt eine schematische Sicht der Anwendung.

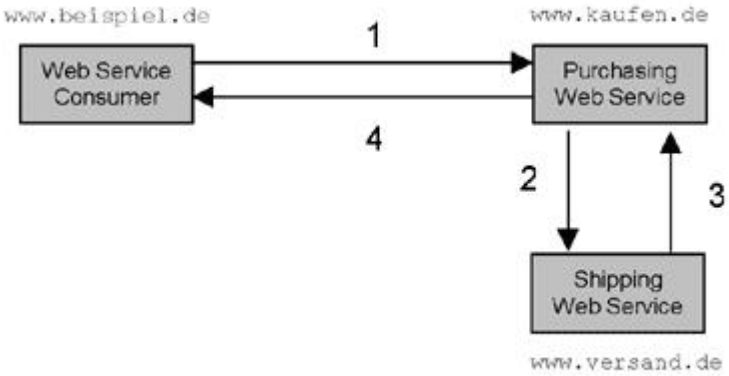


Abbildung 5.7: Beispiel für mehrstufige Webservices

1. Der Aufrufer eines Webservices erteilt einen Kaufauftrag via HTML-Formular. Sobald das Formular gesendet ist, wandelt eine lokale Anwendung das Formular in

ein XML-Dokument um und integriert dieses in den *Body* einer SOAP-Nachricht. Weiterhin werden von der Anwendung die entsprechenden Benutzerinformationen in den WS-Security-Teil des SOAP-*Envelope-Headers* eingefügt, entweder unter Zuhilfenahme klassischer Authentifizierungsmechanismen oder einer (signierten) SAML *Assertion*. Sobald die SOAP-Nachricht erstellt ist, wird diese von der Anwendung via HTTPS übertragen.

2. Der den Kauf verbuchende Webservice erhält die SOAP-Nachricht von dem Aufrufer des Webservice, entschlüsselt diese und verarbeitet die in dem WS-Security-Element enthaltenen Sicherheitsinformationen des SOAP-*Envelope-Headers*. Ist die Entschlüsselung erfolgreich, hat sich der Benutzer dadurch authentifiziert. Anschließend wird die im *Body* des SOAP-Dokuments befindliche Information analysiert, um die Gesamthöhe des Kaufauftrags zu ermitteln. Die in den Preisinformationen enthaltenden Elemente werden lokalisiert und zusammengerechnet, um so das Gesamtvolumen der Kauf-Order zu ermitteln. Der den Kauf verbuchende Webservice vergleicht das Gesamtvolumen des Auftrags mit den Berechtigungen des Käufers, um sicherzugehen, dass dieser zu einer Transaktion in dieser Höhe befugt ist. Bei erfolgter Autorisierung wird eine Anfrage an den für die Auslieferung zuständigen Webservice gestartet, womit der Webservice für die Kauf-Order jetzt als Anfragensteller eines weiteren Webservices agiert. Diese Anfrage wird wiederum in eine SOAP-Nachricht eingebunden, die Sicherheitsinformationen im SOAP-*Envelope-Header* sowie eventuell weitere Daten im SOAP-Envelope-Body enthält.

3. Sobald der für den Versand der Ware zuständige Webservice diesen Versandvorgang erfolgreich ausgelöst hat, schickt er eine SOAP-Nachricht an den Webservice für die Kauf-Order.

4. Der für die Kauf-Order zuständige Webservice (oder alternativ der Versand-Webservice) informiert den Käufer, dass der Kaufvorgang erfolgreich abgewickelt wurde und die Ware ausgeliefert wird. [21]

### 5.3.2 Projekte aus der Praxis

In diesem Abschnitt werden nun im Gegensatz zum vorangegangenen Abschnitt konkrete Anwendungen der WS-*-Spezifikation analysiert. Zum Einen wird das Sicherheitsinfrastruktur des Globus Toolkits aus dem Bereich *Grid-Computing* betrachtet und schließlich ein Projekt aus der Industrie: Die Absicherung des Logistiksystems des Automobilherstellers Volkswagen.

#### GSI im Globus Toolkit (*Grid-Computing*)

*Grid-Computing* bezeichnet alle Methoden, die Rechenleistung vieler Computer innerhalb eines Netzwerks so zusammenzufassen, dass über den reinen Datenaustausch hinaus die (parallele) Lösung von rechenintensiven Problemen ermöglicht wird (verteiltes Rechnen). Jeder Computer in dem „Gitter“ ist eine, den anderen Computern gleichgestellte Einheit.

Damit kann, zu deutlich geringeren Kosten, sowohl die Kapazität als auch die Rechenleistung heutiger Supercomputer übertroffen werden.
Die Globus-Allianz ist ein Forschungs- und Entwicklungsprojekt, um Grid-Anwendungen im Ingenieur- und wissenschaftlichen Bereich voranzutreiben. Das Toolkit enthält viele Services und praktische Komponenten, die man einzeln oder zusammen verwenden kann, um Grids zu bauen und Anwendungen zu programmieren. Entwickelt wurde der Globus Toolkit von Ian Foster und Carl Kesselman. Es ist ein integriertes Toolkit für die Entwicklung von *Grid*-Anwendungen.[22]
Die *Globus Security Infrastructure* (GSI) ist ein wesentlicher Bestandteil des Globus Toolkit. Das Ziel der GSI ist es, eine sichere Kommunikation zwischen den einzelnen Elementen des *Grids* zu ermöglichen. Da die Kommunikation dabei auch organisationsübergreifend erfolgen soll, ist kein zentral verwaltetes Sicherheitssystem möglich. Um die Verwendung des Grids für die Nutzer möglichst einfach zu gestalten, ist ein *Single Sign-On*-Mechanismus (SSO) vorgesehen. Die GSI bietet Dienste für sichere Authentifizierung und Kommunikation über öffentliche Netzwerke. Diese Dienste können von Anwendungen und anderen Komponeneten des Globus Toolkits verwendet werden.[23]
GSI ist dafür zuständig die Identität von Benutzern oder Diensten zu ermitteln (Authentifizierung), die Kommunikation zu schützen und festzustellen, wem es erlaubt ist, welche Aktion auszuführen (Autorisierung). Weiterhin verwaltet das GSI Benutzerinformationen sowie Informationen über Gruppenzugehörigkeiten. Das Globus Toolkit unterstützt unter anderem auch Methoden zur Authentifizierung und Autorisierung aus dem Bereich WS-Security. Diese umfassen:

- Sicherheitsmechanismen auf Nachrichtenebene: Sie implementieren die WS-Security und die WS-SecureConversation Spezifikation, um den SOAP-Nachrichtenverkehr zu schützen.
- Sicherheitsmechanismen auf Transportebene: Sie benutzen sichere Transportmechanismen, wie *Transport-Level Security* (TLS) bzw. *Secure Socket Layer* (SSL)
- Rahmenwerk zur Authentifizierung: Erlaubt über eine Vielzahl von Authentifizierungsmöglichkeiten, den Zugriff auf ein *Grid* mittels dem SAML Protokoll (*Secure Assertion Markup Language*).

Im folgenden werden nun die *Message-Level*-Sicherheitsmechanismen näher untersucht: Zunächst wird die sicherheitsrelevante Verarbeitung von SOAP-Nachrichten auf dem Server betrachtet. Die folgende Abbildung veranschaulicht die Anordnung der betroffenen *Handler* auf einem Server.

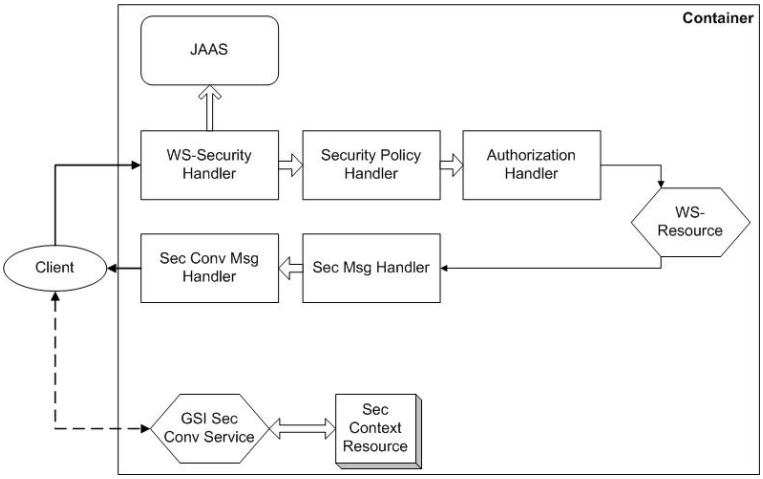


Abbildung 5.8: Sicherheitsrelevante Handler auf einem Server

Es werden zwei unterschiedliche Methoden zur Authentifizierung angeboten (angedeutet durch die zwei unterschiedlichen Linien): GSI *Secure Conversation* und GSI *Secure Message*. Auf diese Möglichkeiten wird hier jedoch nicht weiter eingegangen, da sie eigentlich Bestandteil des Authentifizierungs-Framework sind.
Hat nun die Authentifizierung auf einen der beiden Wege stattgefunden und eine SOAP-Nachricht erreicht den Server, werden verschiedene sicherheitsrelevante *Handler* ausgeführt. Der Erste, der WS-Security *Handler*, durchsucht die Nachricht nach WS-Security *Header*. Aus diesen *Headern* werden dann sicherheitsrelevante Informationen, wie beispielsweise X.509-Zertifikate oder eine Referenz zu einer bereits bestehenden *Secure Conversation*-Sitzung extrahiert. Weiterhin werden alle vorhandenen Signaturen überprüft und Elemente aus dem SOAP-Rumpf decodiert. Der *Handler* erzeugt nun ein gleichwertiges JAAS-Objekt (Java Authentication and Authorization Service) mit Informationen aus der Überprüfungs- und Dechiffrierphase.
Der nächste *Handler* ist der *Security Policy Handler*. Dieser stellt sicher das eingehende Nachrichten den eventuellen Sicherheitsrichtlinien der Dienstressource erfüllt. Diese Richtlinien sind in einen sogenannten *Security Deskriptor* beschrieben.
Dem *Security Policy Handler* folgt der *Authorization Handler*. Dieser verifiziert, ob der Sender der Nachricht autorisiert ist, den entsprechenden Webservice zu konsumieren.
Nachdem die Nachricht den *Authorization Handler* passiert hat, können noch weitere nicht-sicherheitsrelevante *Handler* folgen, die hier aber nicht betrachtet werden. Die Antwort des Webservice passiert dann zwei weitere *Handler*, die in Zusammenhang mit der erfolgten Authentifizierung stehen und für die Autorisierung nicht relevant sind.
Nachdem nun die Server-seitigen Sicherheitsmechanismen beleuchtet wurden, wird nun die Verarbeitung von SOAP-Nachrichten auf dem Client näher betrachtet. Die folgende Abbildung veranschaulicht die Anordnung der betroffenen *Handler* auf einem Client.

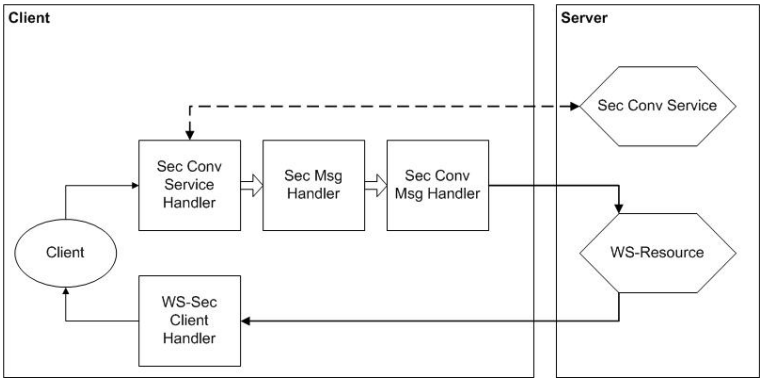


Abbildung 5.9: Sicherheitsrelevante Handler auf einem Client

Die Client-Anwendung kann analog zur Authentifizierung der SOAP-Engine aus zwei Alternativen wählen: GSI *Secure Conversation* oder GSI **Secure Message**. Es gibt drei *Handler* für ausgehende Nachrichten, die entsprechend der beiden Alternativen die SOAP-Nachrichten formatiert und versendet.
Der *Handler* für eingehende Nachrichten heißt WS-Security *Client Handler* und ist zuständig für die Verifizierung und Dechiffrierung von signierten und codierten SOAP-Nachrichten.[24]
Somit ist es also möglich, Grid-Anwendungen, die mit dem Globus Toolkit entwickelt wurden und SOAP zur Kommunikation nutzen, mit Hilfe von Sicherheitsstandards für Webservices abzusichern.

### *SOAP Content Inspector* sichert das VW-Logistiksystem

Mit seinen rund 15 auf der Welt verstreuten Fertigungsstandorten hat Volkswagen Tag für Tag eine Fülle logistischer Herausforderungen zu bewältigen. Komplexe Beschaffungs- und Transportprozesse versorgen die Produktionsstätten mit den benötigten Teilen. Doch die Lieferströme haben sich dramatisch verändert, nämlich von der sternförmigen Verteilung zur Vernetzung.

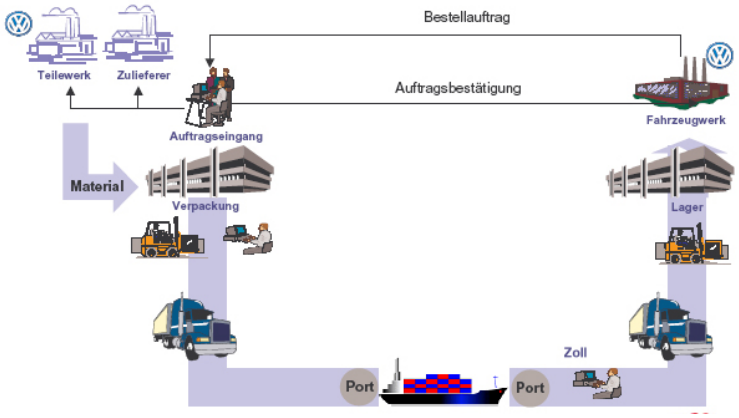


Abbildung 5.10: Ausgangssituation des VW Logistiksystems

Eines neues System zur Überwachung der noch komplexer und komplizierter gewordenen Materialströme in der VW-Welt trägt den Namen GLOBUSS (*GLOBal Unit Supply Survey*). So wissen die Verantwortlichen an den Fertigungsstätten von VW immer exakt, wo sich bestimmte Teile gerade befinden und wann sie mit der Lieferung rechnen können. Voraussetzung ist allerdings, dass diese Verantwortlichen überall auf GLOBUSS zugreifen können. GLOBUSS läuft im Intranet von VW und dieses steht im wesentlichen den externen Logistik-Dienstleister nicht offen. Die naheliegende Lösung ist, diesen Dienstleister einen abgesicherten Zugang über das öffentliche Internet zu verschaffen. Doch konventionelle Firewalls waren keine Lösung, da die Gefahr von „Schlupflöchern“ für Angreifer besteht und herkömmliche Firewall-Lösungen am speziellen Datenprotokoll von GLOBUSS scheitern.

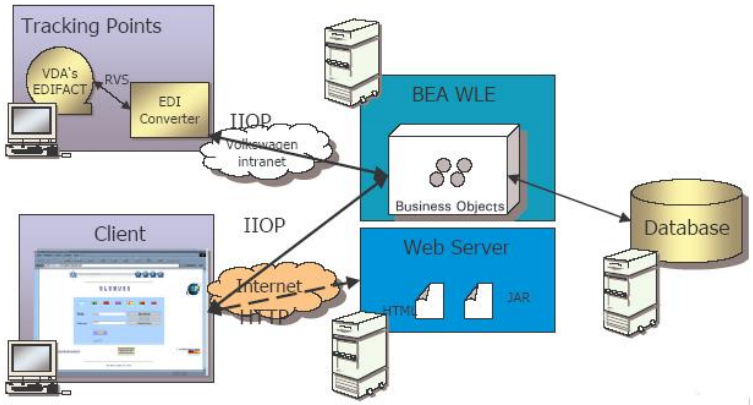


Abbildung 5.11: Basisarchitektur von GLOBUSS

Die Lösung für das GLOBUSS-Problem bestand in einem Sicherheitssystem mit Namen *Domain Boundary Controller* (DBC), das einerseits den notwendigen Informationsfluss zwischen Intranet und Internet ermöglicht, andererseits aber auch zuverlässig vor einem nicht autorisierten Zugriff schützt. Das DBC ist ein Softwaresystem aus den Bereich *Enterprise Application Integration* (EAI) und wurde in der ursprünglichen Version mit EAI-üblichen Standards wie CORBA oder RMI realisiert.
Volkswagen nutzt neben diversen klassischen DBCs auch eine erweiterte Version, genannt **SOAP Content Inspector** oder WS-DBC, um die Internetverbindung zu seinen Zulieferern abzusichern. WS-DBC basiert auf Webservices und unterstützt mehrere offene Standards, wie WS-Security, SAML und XACML, um diese optimal abzusichern.[25]
Der *Web Services Domain Boundary Controller* wird daher auch als XML/SOAP-Firewall bzw. als WS-Security Gateways bezeichnet. Das System versteckt die eigentlichen Webservices hinter virtuellen *Service Endpoints* und inspiziert alle SOAP-Nachrichten, wobei Nachrichten mit inkorrekten oder bösartigen Inhalt blockiert werden. In Zusammenarbeit mit der SAML und XACML stellt der WS-DBC eine hochentwickelte Instanz zur Durchsetzung von Sicherheitsrichtlinen dar, der transparent und ohne Modifikationen in eine bestehende Software-Architektur integriert werden kann.[26]
Wie der *SOAP Content Inspector* nun tatsächlich im VW-System eingesetzt ist, um die die entsprechenden Webservices zu schützen ist in der folgenden Abbildung zu sehen:

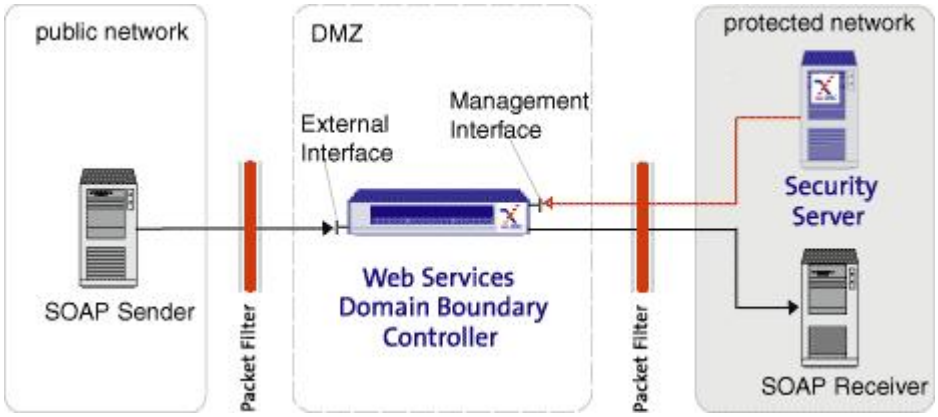


Abbildung 5.12: Absicherung durch WS-DBC

Es werden zwei herkömmliche Firewalls benutzt, um ein Subnetz oder eine sogenannte Demilitarisierte Zone (DMZ) zu errichten. In der DMZ befindet sich der WS-DBC, der das eigentliche Netz schützt, indem der eingehende SOAP-Nachrichtenverkehr analysiert und eventuell unterbrochen wird. Der SOAP Empfänger kann dann eine alleinstehende Anwendung, ein WebContainer oder ein Application Server sein. Die Firewalls der DMZ sind dafür zuständig zu verhindern, das unbefugt auf das WS-DBC von außerhalb oder innerhalb der geschützten Domäne zugegriffen wird. In diesem Fall befindet sich ein *Security Policy Server* innerhalb des geschützten Bereichs, der durch Synchronisation mit dem *SOAP Content Inspector* und anderen DBCs des Gesamtsystems einheitliche Sicherheitsrichtlinien zentral durchsetzt.[27]

### Schlussfolgerungen

Sicherheit ist die letzte große Hürde, die es bei den Webservices noch zu nehmen gilt, bevor die Integrationstechnologie zum endgültigen Durchbruch im Bereich der kommerziellen Business-Anwendungen ansetzen kann. Mit den momentan verfügbaren Technologien und Standards sind die Einsatzmöglichkeiten der Webservices über die Grenzen eines Unternehmens hinweg relativ beschränkt. Dies dürfte sich in naher Zukunft aber ändern: Die meisten der notwendigen Spezifikationen, wie WS-Policy, mit der die Sicherheitsrichtlinien eines Webservices definiert werden, WS-SecureConversation für verschlüsselte Datenverbindungen zwischen Webservices oder WS-Federation für die Herstellung von Vertrauensbeziehungen, befinden sich in der Standardisierungsphase und werden in der nächsten Zeit verfügbar werden.[28]
Der Verbreitungsgrad von Webservices nimmt aber jetzt schon immer mehr zu und gleichzeitig entwickeln sich die damit verbundenen Sicherheitsstandards kontinuierlich weiter. Unternehmen, die nicht rechtzeitig Strategien für Webservices und entsprechende Sicherheitsmaßnahmen entwickeln, werden anfällig für Attacken. Die Durchsetzung tragfähiger Standards hilft, die Sicherheitsbedenken in den Unternehmen zu verringern.
Viele Unternehmen haben daher vor, ihren Webservices-Implementierungen eben diese offenen Sicherheitsstandards zugrunde zu legen. Vernünftigerweise wird ebenfalls in vielen Firmen geplant, mehrere Standards in Kombination einzusetzen, um maximale Sicherheit zu erlangen.[29]

# Literaturverzeichnis


[1] „Sun J2EE (Servlets, JSP, EJB) & Sun One“, B. Sc. Alexander Ernst, http://wwwbayer.in.tum.de/lehre/SS2003/HSEM-bayer/Ausarbeitung9.pdf

[2] „WikiPedia, die freie Enzyklopädie“, http://de.wikipedia.org/wiki/Apache_Axis

[3] „Apache AXIS Architektur“, J.M. Joller, http://www.joller-voss.ch/apache/axis/notes/ApacheAxisArchitektur.pdf

[4] „WikiPedia, die freie Enzyklopädie“, http://de.wikipedia.org/wiki/Application_Server

[5] „Software-Kompetenz.de“, http://www.software-kompetenz.de/?9941

[6] „Apache AXIS - Web Services“, http://ws.apache.org/axis/java/user-guide.html

[7] „Microsoft .NET Austria“, http://www1.microsoft.at/net/story.aspx?id=6775

[8] „MSDN Home“, http://msdn.microsoft.com/library/deu/default.asp?url=/library/DEU/vbcon/html/vbconIntroductionToWebFormsSecurity.asp

[9] „Web Services 101 in ASP.NET“, Christoph Wille, http://www.aspheute.com/artikel/20010621.htm

[10] „PEAR - PHP Extension and Application Repository“, http://pear.php.net/

[11] „WikiPedia, die freie Enzyklopädie“, http://en.wikipedia.org/wiki/JWSDP

[12] „XML, DOM, SAX, JAXP und JDOM mit Java“, http://www.torsten-horn.de/techdocs/java-xml.htm

[13] „XML Magazin und Web Services“, http://www.xmlmagazin.de/itr/online_artikel/psecom,id,101,nodeid,69.html

[14] „Sun Developer Network“, http://java.sun.com/webservices/downloads/webservicespack.html

[15] „Internetnews.com“, http://www.internetnews.com/dev-news/article.php/1555291

[16] „VeriSign Deutschland“, http://www.verisign.de/products-services/security-services/ pki/xml-trust-services/

[17] „MSDN Deutschland“, http://msdn.microsoft.com/library/deu/default.asp?url=/library/deu/dntaloc/html/websvcs_usingsoap.asp

[18] „MSDN Deutschland“, http://msdn.microsoft.com/library/default.asp?url=/
library/en-us/wse/html/f52a7d75-47c0-4d91-af98-6dd4895fa6b3.asp

[19] „MSDN Home“, http://msdn.microsoft.com/library/default.asp?url=/
library/en-us/wse/html/f52a7d75-47c0-4d91-af98-6dd4895fa6b3.asp

[20] „MSDN Home“, http://msdn.microsoft.com/library/default.asp?url=/
library/en-us/dnwssecur/html/securitywhitepaper.asp

[21] „XML Magazine und Webservices“,
http://www.xmlmagazin.de/itr/online_artikel/psecom, id,468,nodeid,69.html

[22] „WikiPedia, die freie Enzyklopädie“,
http://de.wikipedia.org/wiki/Grid#Grid-Projekte

[23] „Grid Computing“, R.Irniger,
http://www.ifi.unizh.ch/richter/Classes/sem_cutting_edge/
summary_Gridcomputing.pdf

[24] „The Globus Toolkit“, http://www.globus.org/toolkit/docs/4.0/security/

[25] „GLOBUSS Anwenderbericht“, PrismTech Limited
http://www.xtradyne.com/documents/collateral/
Anwenderbericht_Volkswagen_Xtradyne.pdf

[26] „Xtradyne Technologies“, http://www.xtradyne.com/products/ws-dbc/ws-dbc.htm

[27] „Xtradyne Technologies“,
http://www.xtradyne.com/products/ws-dbc/WSDBCarchitecture.htm

[28] „Web Services: Langsam, aber sicher“, Urs Bertschy
http://www.innovativetimes.ch/Default.aspx?tabid=121

[29] „Netegrity Studie“,
http://www.contentmanager.de/magazin/news_h6299_netegrity-studie
_unternehmen_noch_zurueckhaltend.html

# Kapitel 6

# Intrusion Detection und Intrusion Prevention

*Lars Biermanski*

*Unternehmen sind zunehmend auf die Integration ihrer Netzwerke an das Internet angewiesen. Dies erfordert aber auch gleichzeitig eine effektive Sicherung der eigenen Systeme. Die Vergangenheit hat gezeigt, dass einfache Paketfilter nicht genügend zum Schutz vor Angriffen beitragen. Dieses Kapitel beschäftigt sich mit neueren Sicherheitskonzepten, die den Administrator bei seiner Arbeit unterstützen oder sogar eigenständig Gegenmaßnahmen bei einem Angriff ergreifen können. Intrusion Detection und Prevention Systeme werden derzeit intensiv entwickelt. Es gibt sie in zahlreichen Variationen für unterschiedliche Anwendungszwecke. Dennoch lassen sich die verschiedenen Ansätze allgemein in Kategorien einteilen. Dieses Kapitel erstellt ein abgerundetes Bild über die Thematik.*

## Inhaltsverzeichnis

## 6.1 Einführung

Das Internet läutete eine neue Ära in der Computerbranche ein. Die zunehmende Vernetzung eröffnete viele neue Möglichkeiten, von denen jeder profitieren sollte. Doch wo Licht ist, ist auch Schatten. Die Anbindung ans Internet bedeutet nicht nur, dass man aus einem überaus großen Informationspool schöpfen kann, der beteiligte Rechner wird selbst Teil des Internets. Besonders für Firmen und Unternehmen ist dies in Hinblick zur Industriespionage äußerst bedenklich.

Zwar möchte man im Regelfall eigene Informationen wie eine Webpräsenz oder einen Online-Shop zur Verfügung stellen, doch sensitive Daten bzw. nicht für die Öffentlichkeit bestimmte sollen auch nicht erreichbar bleiben.

### 6.1.1 Angriffsmöglichekiten

Die Vergangenheit hat gezeigt, welche bedeutende Rolle Sicherheitslücken in Anwendungen und Betriebssystemen einnehmen. Dabei muss die Software nicht unbedingt schlecht oder fehlerhaft programmiert worden sein. Methoden oder Konzepte werden nur von Angreifern in einer Art und Weise genutzt, an die die Entwickler der Software nicht gedacht haben.

Generell können Angriffe in drei Kategorien eingeordnet werden. Normalerweise beginnt eine organisierte Attacke mit einer Aufklärungsphase (Reconnaissance-Phase). Ziel ist es Informationen über das zu kompromittierende System zu gewinnen. Typischerweise geschieht dies durch TCP/UDP-Port Scans oder Brut-Force Passwort Angriffe.

Der nächste Schritt ist der Versuch eines Exploits. Dabei versucht der Angreifende Schwachstellen oder versteckte Funktionen einer Anwendung zu seinem Vorteil zu nutzen um z.B. volle root-Rechte auf den System zu erlangen.

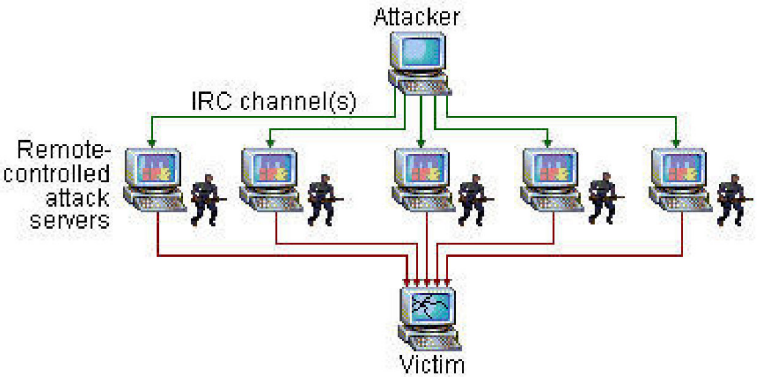


Abbildung 6.1: Beispiel eines DDoS-Angriffes

DoS oder DDoS-Angriffe haben einen destruktiven Charakter. Der Angreifer möchte in der Regel keine Informationen gewinnen, sondern das System oder den Service durch Füllen der Festplatte, Überlasten der CPU oder der Netzwerklinks ausschalten. Um dies

zu erreichen benötigt man normalerweise sehr viele Computer, die zahlreiche Anfragen an das Zielsystem richten. Diese Rechner werden zumeist nicht freiwillig zur Verfügung gestellt, sonder mit einem Wurm, Trojaner oder ähnlichem gekapert. Abbildung 6.1 zeigt einen typischen DDoS-Angriff. Der Angreifer kommuniziert über das IRC-Protokoll mit seinen eingeschleusten Programmen und initiiert die Attacke. Dabei sichert der Angreifer insofern ab, als dass er keine direkten Spuren beim Zielsystem hinterlässt.

Um der ungehinderten Ausnutzung solcher Sicherheitslücken entgegenzuwirken, hatte man das Konzept des Paketfilters bzw. der Firewall eingeführt. Anhand der Quell- und Zielports bzw. Quell- und Ziel-IP-Adressen konnte man nun die Pakete sortieren und gegebenenfalls verwerfen. Andernfalls wird es in das lokale Netzwerk weitergeleitet.

### 6.1.2 Defizite

Das Problem wurde mit der Firewall allerdings nur teilweise gelöst. Pakete mit bösartigen Inhalt, die über legitime Ports an der Firewall ankommen, werden, wie in Abb. 6.2 gezeigt, anstandslos weitergeleitet. Schaden könnte das bei Anwendungen mit Sicherheitslücken, wie ein Webbrowser oder ein e-Mail-Client.

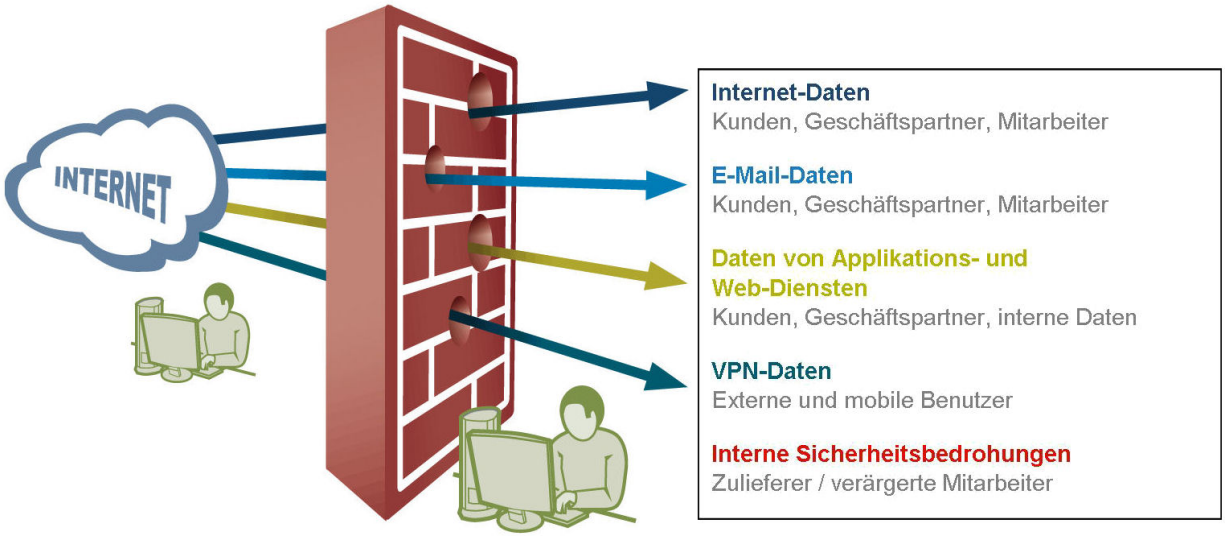


Abbildung 6.2: Die Firewall ist durchlässig [4]

Dabei muss der Angriff nicht unbedingt gegen Schwächen der Software sein, sie könnte sich auch auf den Faktor Mensch verlassen. Ein Beispiel sind e-Mail-Anhänge, über die heute noch immer Rechner kompromittiert werden können, obwohl das Problem schon seit geraumer Zeit bekannt ist.

Hat ein Angreifer einen Weg in ein Rechensystem gefunden, wird er in der Regel versuchen, auf diesen ein Exploit wie so genannte Rootkits zu laden und auszuführen mit dem Ziel, alle Rechte über das System zu erlangen oder es für seine eigene Zwecke zu verwenden.

So kann das kompromittierte System vom Angreifer vielfältig missbraucht werden. Oftmals wird das System selbst Ausgangspunkt eines erneuten Angriffs, es werden Daten

gestohlen oder zerstört. Ein erfolgter Angriff muss allerdings nicht unbedingt ersichtlich sein. Statt die Webpräsenz auf dem Webserver auszutauschen, können auch nur Passwort-Dateien ausgelesen oder Hintertüren installiert werden.

Die Angriffsmöglichkeiten von außerhalb des eigenen Netzes sind äußerst vielfältig und die Firewall bietet eine sehr gute Unterstützung. Allerdings kann der Aggressor auch unmittelbar im eigenen Netz sitzen. Dies können z.B. verärgerte Mitarbeiter sein. Gegen diese Form von Angriffen ist eine Firewall konstruktionsbedingt vollkommen unwirksam.

Um Angriffe überhaupt zu erkennen, müssen Administratoren die Protokolldateien der Firewalls und Betriebssysteme regelmäßig überprüfen. Ein Problem bei dieser Prüfung stellen das Datenvolumen und die Interpretationsgabe des Administrators dar.

Bei der Fülle von Log-Einträgen ist es für einen Menschen nicht einfach einen potenziellen Angriff rechtzeitig zu erkennen. Hinzu kommt, dass die Anzahl der vernetzten Geräte stetig wächst und somit zu den wachsenden Protokolldaten auch der administrative Aufwand wächst. Im Schnitt werden für die eingesetzte Software zwei bis drei Sicherheitsupdates veröffentlicht. Eine Aktualisierung würde also Zeit in Anspruch nehmen, in der die Rechner gegen diese bekannte Sicherheitslücke verwundbar sind.

## 6.2 Intrusion Detection

Zeit verschaffen ist eine Aufgabe eines Intrusion Detection Systems (IDS). Dieses Konzept versucht den Administrator so gut wie möglich zu unterstützen. Es untersucht den anfallenden Datenverkehr genau und erkennt Vorgänge, die vom Administrator nicht gestattet wurden. Die gewonnenen Informationen werden gespeichert und das IDS schlägt gegebenenfalls Alarm.

Ein reines IDS verhält sich ansonsten passiv und macht sich durch nichts bemerkbar. Mit einem gut konfigurierten IDS können Angriffe schon entdeckt werden, während sie stattfinden. Normalerweise findet vor einem tatsächlichen Angriff auch eine Reconnaissance-Phase statt.

Ein IDS könnte dieses Aufklärungsverhalten frühzeitig erkennen und somit bei der Beseitigung der betroffenen Sicherheitslücken helfen. Die Schwierigkeit eines IDS ist die korrekte Festlegung der Kriterien, nach der das System entscheidet, ob es sich um ein Angriff handelt oder nicht. Oftmals sind die vielfältigen Angriffsmöglichkeiten von dem ungefährlichen und normalen Datenverkehr kaum zu unterscheiden. Zudem muss der Angreifer seine Methode nur minimal ändern um vor dem IDS unerkannt zu bleiben.

Die Wahl der Kriterien spielt also eine entscheidende Rolle und ist sehr aufwändig. Zu allgemein gefasst, können Verhaltensmuster als Attacke interpretiert werden, die gar keine sind. Derartige Fehlalarme werden falsch positiv genannt. Werden die Kriterien allerdings zu eng gefasst, könnten Attacken stattfinden, die das System übersieht. Derartige Vorkommnisse nennt man falsch negativ. Anders als bei Spam-Filtern sind falsch negative Fehler fatal, da genau das passiert, was man mit einem IDS zu verhindern versucht. Die

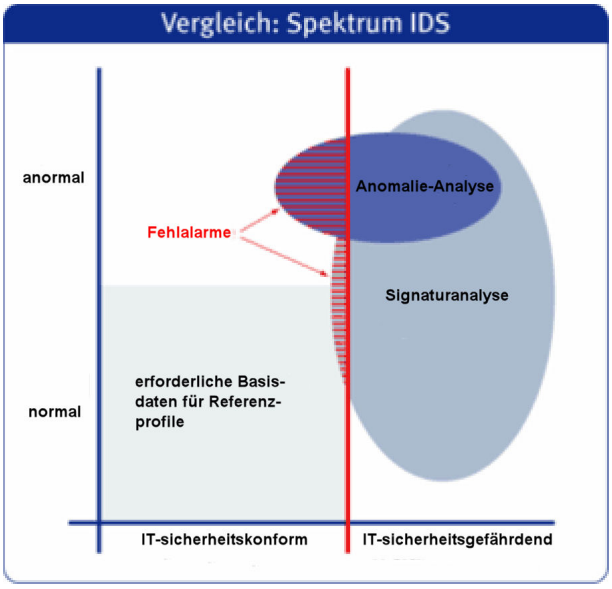


Abbildung 6.3: Meldungen eines IDS [7]

Gefahr in falsch positiven Alarmen liegt darin, dass man durch eine Vielzahl von Fehlalarmen tatsächliche Alarme übersehen könnte. Im Idealfall kann man die Fehlermeldungen, wie in Abb. 6.3 gezeigt, auf ein Minimum beschränken, aber niemals völlig ausschließen. Die in dieser Abbildung erwähnten Analyseverfahren werden in Kap. 6.2.2, S. 111 beschrieben.

Intrusion Detection Systeme gibt es für unterschiedliche Anwendungszwecke in verschiedenen Variationen. Dabei spielt das zu betrachtende System eine Rolle. IDS gibt es auf Anwendungs-, Host- und Netzwerkebene.

### 6.2.1 Zeitliche Abfolge der Auswertung

Je nachdem zu welchen Zweck das IDS eingesetzt werden soll und welche personellen und hardwaretechnischen Ressourcen zur Verfügung stehen, fällt die Wahl entweder auf die Echtzeitauswertung oder die Batch- oder Intervallorientierte Auswertung.

#### Batch- oder Intervallorientierte Auswertung

Letztere Variante dient weniger dem schnellen Aufspüren von stattfindenden Angriffen, als der akribischen Speicherung des Datenverkehrs und der forensischen Beweissicherung. Gerade für die juristischen Praktiken zur Regelung der Strafverfolgung im Bereich der Computerkriminalität ist dieses Verfahren gängig und das vor allem, weil es auch nach der Analyse möglich ist manuell alle Daten nochmals beliebig oft nachzuprüfen.

Weitere Argumente sind die geringere Prozessorlast und die Möglichkeit die Analyse auf ein externes Gerät auszulagern. Allerdings wird je nach zu überwachendem System eine mehr oder weniger große Plattenspeicherplatzmenge benötigt, da die unverarbeiteten

Daten bis zur Analyse und vielleicht darüber hinaus gespeichert werden müssen. Findet tatsächlich ein Angriff statt, müssen gerade diese Daten besonders geschützt werden. Ein Angreifer hätte sonst die Möglichkeit das Beweismaterial zu eliminieren.

Das Analyseverfahren ist am sinnvollsten, wenn kein Vollzeit-Sicherheitspersonal unterhalten wird. Die Echtzeit-Analyse allein wäre ansonsten nicht sehr zweckdienlich, da möglicherweise niemand auf den Alarm reagiert. Andererseits kann ein unerlaubter Eingriff in das System schon längst stattgefunden haben, ohne dass das IDS Alarm geben konnte. Demnach gestaltet sich das Einleiten von Gegenmaßnahmen äußerst schwer.

#### Echtzeitauswertung

Ein System, das kontinuierlich Informationen sammelt, Daten auswertet und Bericht erstattet, betreibt eine Echtzeitauswertung. Wie schnell diese vonstatten geht, hängt ganz von dem System selbst ab. Normalerweise läuft die Auswertung im Millisekundenbereich ab, sodass ein nahezu sofortiges Einleiten von Gegenmaßnahmen bei einem Angriff möglich ist.

Muss das IDS allerdings eine überaus große Datenmenge pro Zeiteinheit kontrollieren oder reichen die Hardwareressourcen für die erteilte Aufgabe nicht aus, kann die Verarbeitungsgeschwindigkeit des Systems mehr oder weniger stark abnehmen und auf ein Angriff könnte nicht rechtzeitig reagiert werden.

Generell benötigt ein Echtzeit-IDS viel Prozessorleistung und Speicherplatz. Da ein unerlaubter Eingriff in das System immer ein Sonderfall sein sollte und eine gesonderte Aufmerksamkeit verlangt, hilft die Echtzeitauswertung durch ein schnelles Aufspüren eines Angriffs auch bei der schnellen Wiederherstellung des Regelbetriebs.

Juristisch bietet ein solches System die Möglichkeit, während eines Angriffs Informationen über den Angreifer zu sammeln, die später zur Strafverfolgung des Täters beitragen. Dagegen sollte man das Vorhaben einen Gegenangriff zu starten lieber überdenken. Dies kann juristische Folgen haben oder aber die Aufmerksamkeit des Angreifers auf sich ziehen.

Letztere Variante könnte unvorhersehbare Konsequenzen nach sich ziehen. Das Konfigurieren eines Echtzeit-IDS ist in sofern kritisch, dass es möglicherweise laufend zu falsch positiven Meldungen kommen könnte. Die Gefahr dabei besteht durch das „Übersehen“ eines echten Vorfalls.

### 6.2.2 Formen der Analyse

Neben der zeitlichen Abfolge gibt es auch Unterschiede in der Art der Auswertung. Die verschiedenen Ansätze ergeben sich aus den unterschiedlichen Anwendungsgebieten und wurden schließlich bei bestimmten IDS miteinander kombiniert. Zum anderen entwickelt eine Vielzahl von Firmen aus verschiedenen Branchen wie aus dem Anti-Virus-Lager (z.B. McAfee) oder dem Netzwerkbereich (z.B. Cisco) solche Systeme. Dabei wurden bekannte Methoden auf das neue Gebiet portiert.

Heute gibt es drei Möglichkeiten den Datenverkehr zu analysieren: die Signatur-Analyse, die Anomalie-Analyse und die Integritätsanalyse.

### Signatur-Analyse

Das Prinzip der Signatur-Analyse stammt aus der Anti-Viren-Software. Dabei wird die anfallende Datenmenge nach bestimmten Kriterien (Signaturen) durchsucht. Jedem hypothetischen oder tatsächlich stattgefundenen (bekannten) Angriff wird durch den Hersteller ein eindeutiges Muster zugeordnet, das den Angriff eindeutig identifiziert.

So ist das IDS in der Lage einen solchen Angriff zu erkennen. Ein solches Muster kann sehr primitiv, also durch Vergleich bestimmter Zeichenfolgen, als auch sehr komplex anhand von mathematischen Ausdrücken sein. Findet das IDS keine dieser Signaturen in den vorliegenden Informationen, fährt das System fort und verwirft die zu betrachtende Datenmenge bzw. beachtet sie nicht weiter.

Das ist auch schon das größte Problem des IDS. Es erkennt nur bekannte Angriffsformen. Modifiziert ein Angreifer seine Attacke nur minimal zu einer der IDS unbekannten Form, gibt es keinen Alarm. Als Beispiel sei ein Portscan genannt, der einzelne Ports über Stunden oder Tage hinweg abtastet und das möglicherweise noch über mehrere Source-IP-Adressen statt alle Ports in kürzester Zeit zu scannen. Hier taucht außerdem noch das Problem auf, wann etwas als ein Angriff zu werten ist.

Aber selbst wenn ein Signatur-IDS theoretisch über eine Datenbank mit allen denkbaren Angriffen verfügen würde, könnte es einen Angriff nicht rechtzeitig erkennen. Durch die enorme Größe der Datenbank würde das IDS sehr langsam werden.

### Anomalie-Analyse

Ein anderer Ansatz analysiert das „normale“ Verhalten eines Systems. Das kann ein einzelnes Benutzerverhalten als auch der aufkommende Netzwerkverkehr oder auch ein Gerät sein. Die Statistische oder auch Anomalie-Analyse zeichnet Abweichungen von den Standardprofil des zu betrachtenden Objekts auf.

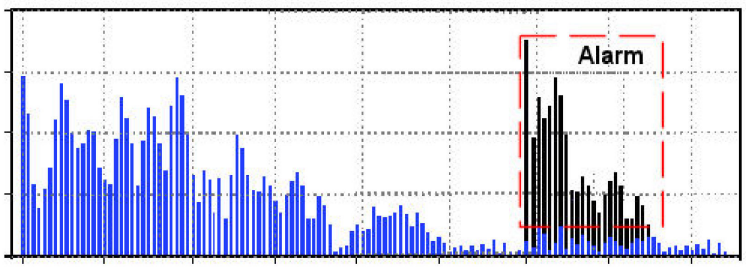


Abbildung 6.4: Erkennung eines Anomalieverhaltens

Dieses Profil wird über einen längeren Zeitraum hin nach typischen Attributen wie Tageszeit der Nutzung, Zugriffe pro Zeiteinheit und wohin mit einer gewissen Toleranz erstellt.

Fällt das Verhalten plötzlich aus dem Muster, wird Alarm gegeben. Dies geschieht z.B. wenn ein Benutzer sich um 4:00 Uhr nachts anmeldet, obwohl er normalerweise zwischen 7:00 und 17:00 Uhr arbeitet. Abb. 6.4 zeigt das „normales“ Verhalten am System (blau) nach der Zeit und eine plötzliche Änderung am Systemverhalten (schwarz). Das IDS würde in diesem Fall Alarm schlagen.

Wird aber das Nutzerverhalten des Objekts ständig und plötzlich verändert, wird das IDS dementsprechend viele falsch positive Meldungen ausgeben. Deshalb wird sich das IDS oft langsam an neue oder andersartige Verhaltensweisen anpassen. Ist das dem Angreifer bekannt, so kann er versuchen über einen längeren Zeitraum hinweg das IDS sozusagen zu seinem Angriff hinzuführen, sodass es diesen nicht als Attacke erkennt.

Ansonsten fällt jede Unregelmäßigkeit auf. Das können bekannte und unbekannte Angriffe oder auch Benutzer des Systems sein, die beispielsweise auf nicht für sie bestimmte Informationen zugreifen wollen.

**Integritätsanalyse**

Die letzte Möglichkeit überprüft Dateien auf einem Computersystem, ob sie seit dem letzten Integritätscheck verändert wurden. Hierzu werden zunächst Hashwerte von jeder zu prüfenden Datei angefertigt und gespeichert.

Läuft die Analyse, werden Hashwerte der Dateien angefertigt und mit den gespeicherten verglichen. Wenn sie sich unterscheiden, wurde die entsprechende Datei verändert und das IDS reagiert. Es ist auch möglich gewisse Attribute der Dateien oder Verzeichnisse wie Zeitpunkt der letzten Änderung oder Dateigröße prüfen zu lassen. So kann das IDS z.B. anschlagen, wenn Log-Dateien kleiner werden.

Überhaupt lässt sich das System sehr fein und individuell konfigurieren. Diese Anpassung muss allerdings meist für jeden Rechner einzeln stattfinden. Wichtig ist die Hashwerte gesondert zu schützen. Ein Angreifer könnte diese sonst verändern und somit sein Angriff verstecken. Aber das setzt Kenntnis über ein solches IDS und über die Architektur des Systems voraus.

Normalerweise wird ein Angreifer gerade durch sein versuchtes unauffälliges Verhalten auffallen. Hat er es geschafft das System zu kompromittieren, wird er oftmals versuchen seine Spuren zu verwischen und gleichzeitig sicherstellen wollen, dass er auch weiterhin Zugriff zum System erhält. Dazu muss er das infizierte System zumindest so verändern, dass ein hochgeladenes Programm beim Systemstart gestartet wird oder dass die Log-Dateien keine Spuren seines Eindringens aufweisen.

Ein Dateien-Integritätscheck wird aber solche Maßnahmen sofort erkennen. Dafür benötigt das System während der Laufzeit relativ viele Systemressourcen.

#### DoS-Analyse

Das ID-System untersucht hierbei den Datenverkehr auf Schwellenwertüberschreitungen. Die Schwellenwertgrenze wird zum einen vom Administrator vordefiniert, als auch durch das durchschnittliche Verhalten abgeleitet.

### 6.2.3 Sensoren

Bevor Informationen in Form von Netzwerkpaketen oder Daten analysiert werden können, müssen diese zunächst von einem IDS-Sensor gesammelt und zur Verfügung gestellt werden.

Allgemein befinden sich die Sensoren entweder auf dem Host selbst oder sind über Interfaces an Netzwerksegmente angeschlossen. Man spricht dann entweder von einem Host-basierenden oder von einem Netzwerk-basierenden Sensor.

#### Host-based IDS

Erstere Variante betrachtet nur das lokale System und untersucht es für sich abgeschlossen auf Anzeichen eines Eindringens. Dabei wird auch der Datenverkehr über verschlüsselte Verbindungen berücksichtigt, wenn sie am zu betrachtenden System enden. Über den Sensor kann das IDS jedes Vorgehen auf dem Rechner detailliert protokollieren und somit einen Angriff genau rekonstruieren.

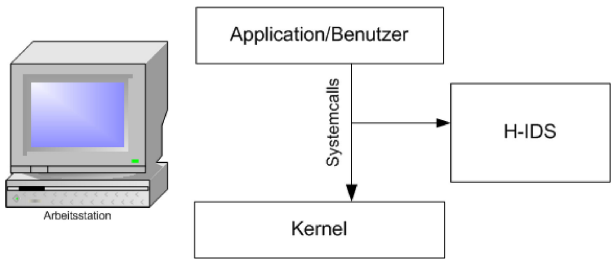


Abbildung 6.5: Ein mögliches H-IDS

Da die Vorgänge auf einem Host eher kalkulierbar sind, produziert das IDS auch weniger falsch positive Meldungen. Die Wirkung von Befehlen und Zugriffen auf den Kernel sind relativ vorhersehbar, während komplexere Angriffe durch die Analyse von Netzwerkpaketen bedeutend schwerer zu erkennen sind.

Jede Instanz muss früher oder später auf den Kernel zugreifen, um für einen Angreifer interessante Funktionen auszuführen. Eine gängige Lösung ist daher, dass das IDS alle Systemcalls mithört, analysiert und gegebenerfalls Alarm schlägt. (Abb. 6.5)

Schlägt ein solches IDS allerdings Alarm, hat ein Angreifer bereits zwangsläufig auf den Host Daten laden können. Erlangt ein Angreifer so rechtzeitig root-Rechte, könnte er das

IDS vor einem Alarm deaktivieren. Ein solcher Fall ist allerdings sehr unwahrscheinlich und fordert Kenntnisse über den vorhandenen Systemaufbau.

Dafür muss ein solches System auch auf jeden Host installiert werden, erfordert somit einen relativ hohen Management-Aufwand und benötigt Systemressourcen des Hosts.

### Network-based IDS

Dagegen wird ein IDS, das über Netzwerk-basierende Sensoren verfügt, lediglich an das Netzwerk angeschlossen. Es findet keine Veränderung an den Hosts selbst statt. Das System erhält einen umfassenden Überblick von der gesamten Netzwerkarchitektur und ist somit in der Lage komplexe und verteilte Angriffe aufzudecken.

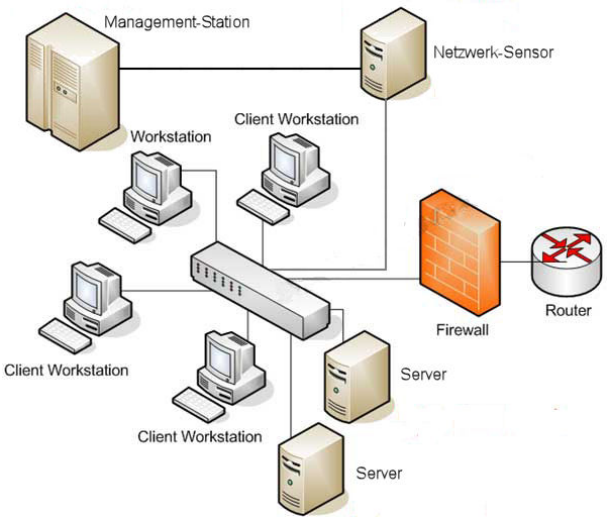


Abbildung 6.6: Typischer Aufbau eines N-IDS

Es besteht aus zwei Komponenten: mindestens ein Netzwerksensor und eine so genannte Management-Station. Die Komponenten sind entweder durch verschlüsselte Verbindungen oder durch ein separates Netzwerk verbunden. Die Management-Station dient meist nur noch als Anzeige- und Kommunikationszentrum.

Die Analyse findet dann direkt auf den Netzwerk-Sensoren statt, um den aufkommenden Datenverkehr zwischen den beiden Einheiten möglichst gering zu halten. Bei einem großen zu überwachenden Netzwerk wäre eine zentrale Analyse auch nicht zweckdienlich, weil eine sehr große Datenmenge zu prüfen wäre.

Der Netzwerk-Sensor ist ein eigenständiger Rechner mit mindestens einem Interface und überwacht mindestens ein Netzwerksegment. Manchmal, vor allem bei kleinen Architekturen werden die beiden Komponenten auch auf einen Rechner zusammengefasst.

Damit der gesamte Datenverkehr aufgenommen werden kann, benötigt der Sensor einen speziellen Netzwerkanschluss in Form eines Mirroring-Ports, wenn es sich um ein geswitchtes Netzwerk handelt. (siehe Abb. 6.6)

Zudem muss sichergestellt werden, dass der Sensor auch bei Vollbelastung des Netzwerkes alle Netzwerkpakete erhält und keine verworfen werden. Das IDS verhält sich im Netzwerk absolut passiv und ist über keine IP-Adresse ansprechbar.

Eine theoretische Schwachstelle ergibt sich lediglich aus der MAC-Adresse, durch die der Rechner direkt ansprechbar ist. Ist dem Angreifer bekannt, um welches IDS es sich handelt, könnte er ebenso versuchen es zu kompromittieren, da jedes Paket in das entsprechende Netzwerksegment auch von dem IDS analysiert wird.

### Network Nodebased IDS

Eine eine Kombination beider Verfahren ist das Network Nodebased IDS. Es wird auf den Host installiert und betrachtet den ein- und ausgehenden Netzwerkverkehr sowie die Zugriffe auf den Kernel des Betriebssystems. Dabei erbt es allerdings alle schlechten Eigenschaften eines Host-basierenden IDS.

### Application-based IDS

Relativ neu auf den Markt sind so genannte Application Based IDS. Sie sind speziell auf bestimmte Anwendungen oder Protokolle abgestimmt und versuchen Anweisungen und Daten an das Programm vorher zu simulieren und zu bewerten.

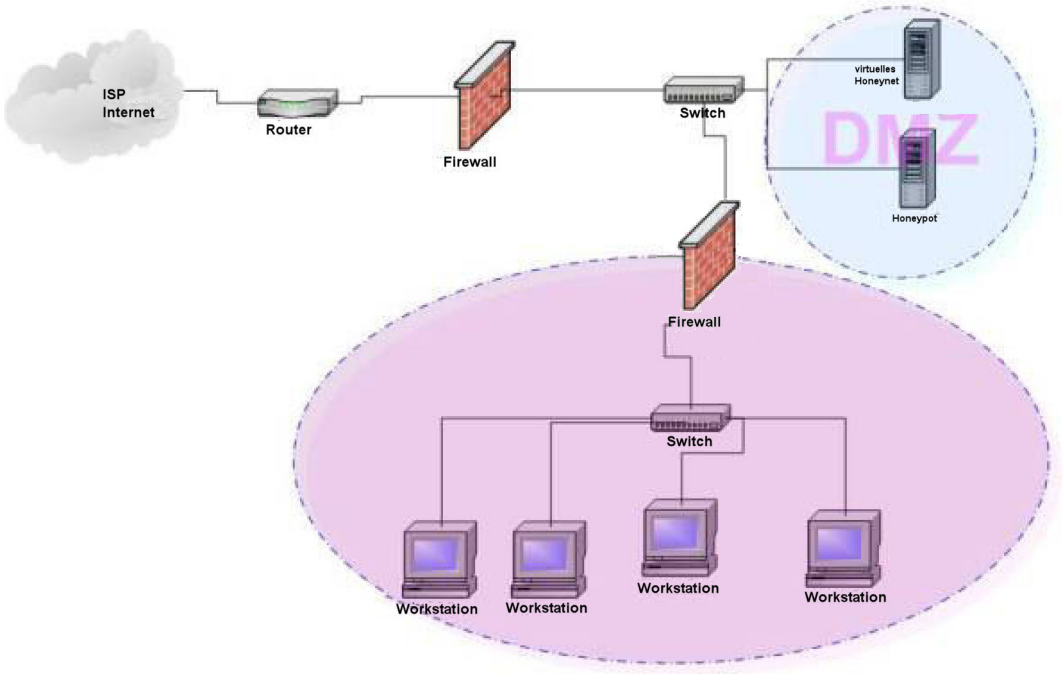


Abbildung 6.7: Aufbau eines Honeypots

Sie arbeiten also im Gegensatz zu den restlichen ID-Systemen auch auf Anwendungsebene. Dabei befindet sich das System entweder auf den Host oder In-Line im Netzwerk.

Ersteres ist in der Lage neben den Netzwerkdaten auch Benutzereingaben und Betriebssystem interne Abläufe zu überprüfen.

Für die In-Line-Version spricht eine Prüfung noch bevor die Daten das Zielsystem erreichen. Gerade bei der Überwachung eines Protokolls bietet sich dieses Verfahren an.

Ein weiteres Konzept bietet einem Angreifer bewusst ein attraktives Ziel, einen Honigtopf (Honeypot), an. Meistens verfügt das angebotende System über bekannte Sicherheitslücken und vermeintlich wichtige Informationen.

Zusätzlich überwachen zahlreiche Sensoren das System und jeder Vorgang wird protokolliert. Da normalerweise nicht auf den Honigtopf zugegriffen wird, sind jede von normalen Netzwerkverkehr abweichende Ereignisse als Angriffe zu bewerten.

Das „echte“ Netzwerk befindet sich parallel zu den Falschsystemen und wird separat geschützt. (Abb. 6.7)

## 6.3 Intrusion Prevention

Ein IDS handelt nur reaktiv und verhält sich passiv. Die Aufgabe eines reinen IDS ist primär die Protokollierung und Berichterstattung. Um einen Angriff abzuwehren, benötigt es also einen Administrator. Die Zahl der Firmen, die ein ganztägiges Sicherheitspersonal für ihre Rechner unterhalten, ist relativ begrenzt.

Das IDS muss also soweit modifiziert werden, dass es vor Angriffen präventiv schützen kann. Das Einleiten automatischer Gegenmaßnahmen beschleunigt den Vorgang durch die Ausschaltung der menschlichen Reaktionsfähigkeit drastisch, sodass das System im Idealfall einen Angriff unmittelbar nach Feststellung eines solchen abwehren könnte.

Das Konzept ist bekannt als *Intrusion Prevention System* oder *Intrusion Protection System*. Manchmal wird auch von *IDS mit Incident Response* gesprochen. Derzeit wird an vielen Firmen intensiv an der Umsetzung und Perfektionierung der Konzepte gearbeitet.

Dabei wird der Begriff *Intrusion Prevention System* für eine derart vielfältige Produktpalette von Sicherheitstechnologien benutzt, dass eine konkrete Beschreibung schwerfällt. Grob formuliert soll ein IPS einen Security-Layer zwischen den zu schützenden Elementen und dem Rest des Systems bilden.

Es gibt IP-Systeme In-Line im Netzwerk, Host-basierend und auf Application-level. Der Unterschied zu den IDS-Varianten liegt im wesentlichen in der Möglichkeit aktiv und automatisch in das Geschehen einzugreifen. Diese Gegenmaßnahmen kann das konkrete Filtern von einzelnen Paketen und Verbindungen sein, als auch das Einleiten juristisch bedenklicher Gegenangriffe. Dabei beschränkt sich das Verteidigungsverhalten bei weitem nicht auf das bloße Verwerfen von Paketen.

Ein IPS kann den Header eines Paketes nahezu beliebig verändern, sodass Pakete mit fragwürdigen Inhalt z.B. nur bestimmte Rechner erreichen dürfen (etwa einen Honeypot).

Aber auch Restriktionen wie die Anzahl von parallelen TCP-Verbindungen oder die Zuweisung einer bestimmten Bandbreite sind problemlos möglich. Es ergeben sich hierdurch auch andere Möglichkeiten für den Administrator, als die reine Verteidigung vor Angriffen. So ist es beispielsweise auch möglich den HTTP-Verkehr von den eigenen Mitarbeitern zu filtern um so das private Surfen am Arbeitsplatz einzudämmen.

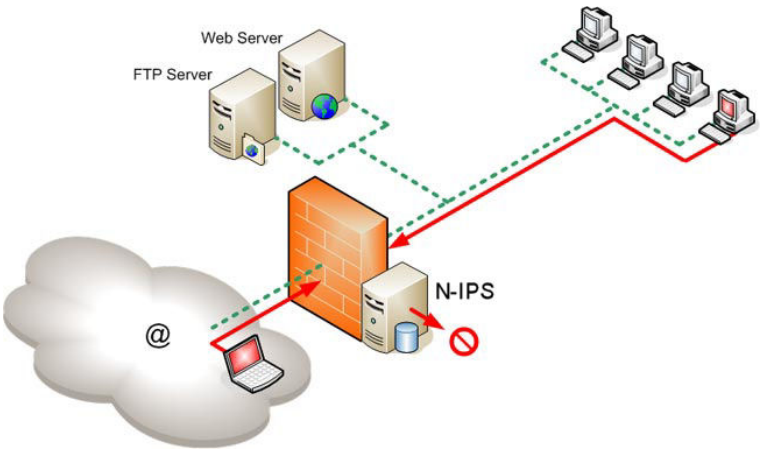


Abbildung 6.8: Aufbau eines N-IPS [8]

Außerdem muss im Netzwerk der gesamte Verkehr – Inline – durch das IPS laufen, bevor es überhaupt weitergeleitet wird. Das Netzwerk IDS dagegen wurde wie ein normaler Endrechner an das Netzwerksegment angeschlossen. Ein N-IPS sitzt also in der Regel direkt hinter einem Paketfilter oder beide Elemente wurden in ein System integriert. (Abb. 6.8)

Dieses Konzept fordert vom IPS eine sehr hohe Verfügbarkeit. Fällt es aus, so wird das betroffende Netzwerksegment vom Rest des Netzwerkes getrennt. Gleichzeitig dürfen sich durch das IPS die Latenzzeit und die Perfomance des Netzwerkes nicht deutlich verschlechtern.

Das Besondere an ein NIPS ist die Lage im Netzwerk. Als Inlinesystem kann es die Pakete am effizientesten filtern. Dabei werden meist verschiedene Techniken bzw. Verfahren mit einander kombiniert um einen möglichst fehlerfreien Betrieb zu gewährleisten. Eine reine Signatur-basierende Lösung wäre nicht zweckmäßig genug. Zunächst wird ein Paket auf seine Fragwürdigkeit geprüft. Bestehen aus Sicht des IPS keine Zweifel, wird das Paket wie bei einer Firewall weitergeleitet. Ansonsten durchläuft das Paket verschiedene Analysevarianten.

Die Kriterien eines IPS sind besonders sorgfältig zu wählen, da jeder falsch positiver Vorfall eine Gegenmaßnahme produzieren könnte, die den regulären Betrieb mehr oder weniger stark stören könnte. Im schlimmsten Fall verärgern die automatischen Gegenmaßnahmen eines solchen Systems die eigenen Kunden oder Mitarbeiter der Firma und verursachen dadurch einen echten finanziellen Schaden.

## 6.4 Schwachstellen

Die Schwachstellen eines IDS oder IPS erfordern meist immer Kenntnisse über das verwendete Programm. Es gibt zwei ähnliche Möglichkeiten, die Schutzprogramme zu umgehen ohne sie zu deaktivieren oder auf sie zuzugreifen.

Dabei wird ein Angriff derart verändert, dass es keinen Alarm geben wird. Im Grunde werden bewusst Pakete zwischen denen mit bösartigen Code eingefügt, die derart beschaffen sind, dass sie vom IDS/IPS akzeptiert, aber nicht vom Zielsystem akzeptiert werden. Man spricht dann von Insertion.

Die andere Technik, Evasion, geht davon aus, dass das IDS/IPS gewisse Pakete ablehnt, die das Zielsystem aber annimmt. Dadurch erhalten die ankommenden Informationen für das Zielsystem und das IDS/IPS unterschiedliche Bedeutung, sodass theoretisch bösartiger Code unerkannt auf dem Host ausgeführt werden kann.

Gerade für Firmen sind kommerzielle Lösungen sehr teuer. Die Anbieter solcher Systeme stellen komplette leistungsstarke Rechner inklusive Signaturupdates zur Verfügung und lassen sich das teuer bezahlen. Typischerweise werden solche Lösungen mit mindestens 10.000 € bezahlt. Freie Lösungen wie Snort, Tripwire oder Shadow sind dagegen Open Source und damit kostengünstig in der Software. Sie erfordern aber einen höheren Installationsaufwand und verfügen über einen geringeren Service. Außerdem muss der Administrator des Systems selbst die erforderliche Hardware besorgen.

## 6.5 Zusammenfassung

ID- und IP-Systeme unterstützen die Arbeit der Administratoren und vervollständigen das Sicherheitskonzept. Vor allem ein vorhandenes IDS ersetzt aber keine Sicherheitselemente wie Zugangskontrollen und Firewalls. Ein IPS dagegen wird IDS und Paketfilter irgendwann vollständig ersetzen.

Der Übergang zwischen den Definitionen von Intrusion Detection und Intrusion Prevention ist sehr fließend und schwammig. Ein reines IDS ist heutzutage nur noch schwer zu finden. Jedes System ermöglicht zumindest einen einfachen TCP-Reset zu senden um die potenziell bösartige Verbindung abzubrechen. So gibt es ID-Systeme mit umfangreichen Möglichkeiten Gegenmaßnahmen einzuleiten, wie auch IP-Systeme mit nur wenigen Abwehrmöglichkeiten.

# Literaturverzeichnis


[1] Gerloni, H., Oberhaitzinger, B.: *Praxisbuch Sicherheit für Linux-Server und -Netze, Hanser, 2004*

[2] Rash, M., Orebaugh, A.: *Intrusion Prevention and Active Response, Syngress, 2005*

[3] Schernthaner, D.: *Security Update Flash, Computacenter, 24.11.2004*

[4] Hofmann, T.: *Intrusion Prevention, McAfee, 18.04.2005*

[5] Götz, C.: *Intrusion Detection Systeme im Vergleich, cirosec, 05.02.2003*

[6] Trinkler, R., Burkhalter,R.: *Intrusion Prevention, 11.02.2004*

[7] Networkers AG *http://www.networkers.de/services/netsec/securitysolutions/ids/, 25.06.2005*

[8] Secudos *http://www.secudos.de/public/Intrusion_prevention.jpg, 25.06.2005*

[9] Top Layer Networks Deutschland *Was kommt nach IDS? Eine Einführung in Network Intrusion Prevention, März 2003*

# Kapitel 7

# Neue Sicherheit im Internet durch IPv6

*Marc Akkermann*

*Das heute im Internet fast ausschließlich verwendete Protokoll IPv4 wurde in den 60er Jahren definiert, in einer Zeit also, als an das Internet nur wenige Teilnehmer angebunden waren, die sich zum größten Teil noch direkt kannten. Dies hat sich grundlegend geändert. Heute sind Hunderte von Millionen von Computern über das Internet vernetzt und die Bedeutung der Daten auf Rechensystemen und der übertragenen Daten wächst täglich. Dies macht nicht nur Spionage interessant, sondern zieht auch die Aufmerksamkeit von Kriminellen und Geltungssüchtigen (z.B. Skript-Kiddys) auf sich. Im Jahre 1995 wurde die nächste Generation des Internet-Protokolls (IPv6) standartisiert. Diese Seminararbeit stellt die Gründe für die Entwicklung des neuen Internetprotokolls genauer dar und gibt einen kurzen Einblick in die wesentlichen Neuerungen. Spezielle Mechanismen wie DHCP, MobileIPv6 etc. werden hierbei außer Acht gelassen, da das Hauptaugenmerk der Arbeit auf den Sicherheitsmechanismen in IPv6 liegt. Nach Beschreibung der Bedrohungen im Netz und der gängigen Gegenmaßnahmen dazu werden dann die speziellen Sicherheitsmechanismen in IPv6 näher erläutert.*

## Inhaltsverzeichnis

## 7.1 Einleitung

Netzwerke haben in den vergangenen Jahren zunehmend an Bedeutung gewonnen. Am deutlichsten wird dies am Beispiel des Internets, welches in den neunziger Jahren einen regelrechten Boom erlebte. Aber auch firmen- und hausinterne Netze werden im alltäglichen Gebrauch immer wichtiger. Ein weiterer Aspekt welcher hinzukam, ist die Mobilität. Netzwerke werden nicht mehr nur von stationären Workstations verwendet, sondern auch von portablen Geräten (z.B. WLAN). Somit müssen Protokolle wie IP vor allem mit folgenden Problemen fertig werden:

- **Die steigende Nutzerzahl**
  Durch die, vor allem im Internet stetig wachsende Zahl an Benutzern steht man in Bereichen wie Adressierung, Routing und Verwaltung vor einer enorm steigenden Komplexität.
- **Die Wegfindung zu mobilen Geräten**
  Routing-Algorithmen müssen für das Aufrechterhalten der Kommunikation zwischen beweglichen Geräten weitaus mehr leisten als in statischen Netzen.
- **Die Sicherheit**
  Durch die steigende Nutzung von Rechnernetzen in allen Bereichen des Lebens und die Mobilität der Endgeräte wird es immer wichtiger die Kommunikation sicher zu gestalten, um Missbrauch vorzubeugen und einen verlässlichen Datenaustausch zu gewährleisten.

Der zuerst genannte Punkt war einer der Hauptgründe für den Start der Entwicklung von IPv6, da man davon ausging, dass der Adressbereich welcher in IPv4 zur Verfügung steht wegen des „Internet-Booms“ bald nicht mehr ausreichen würde. IPv4 wurde ursprünglich für Netze mit einem kleinen Benutzerkreis wie zum Beispiel Hochschulen, Forschungseinrichtungen und Militär entwickelt. Diese Nutzer verwendeten Netzwerke ausschließlich für spezielle Anwendungsdomänen und deswegen hat man bei der Wahl des Adressraums keine zu großen Teilnehmerzahlen berücksichtigt und eine Adresslänge von 32 Bit für ausreichend gehalten. IPv6 bietet eine neue Adressierung mit Adressen der Länge 128 Bit, auf welche im Abschnitt 7.2.1 genauer eingegangen wird.

Die Wegfindung für mobile Geräte wird hier nicht weiter behandelt. Es sei jedoch gesagt, dass in IPv6 effektive Mechanismen für dieses Problem implementiert wurden, welche in IPv4 fehlen. Mehr Informationen diesbezüglich findet man in [1, S. 195-234]

Der Hauptteil dieser Arbeit befasst sich mit den Anforderungen an die Sichereit von IPv6 und den hierzu implementierten Mechanismen. In der Entwicklung von IPv6 wurde dieser Bereich zu einem Paket „IPSec“ zusammengefasst. Da IPv4 noch immer sehr verbreitet ist, wurde IPSec, zumindest zum Teil, auch nachträglich in IPv4 rückportiert. Genaueres hierzu findet sich in Abschnitt 7.5.1.

## 7.2 Grundlagen zu IPv6

In diesem Abschnitt werden kurz die neuen Adressen in IPv6, sowie die Header des Protokolls angesprochen um eine Basis für die Erklärung der weiteren Mechanismen zu legen. Für Informationen über weitere Funktionalitäten wie ICMP, DHCP, Routing etc. sowie Details über die Adressierung und verschiedenen Header wird verwiesen auf [1].

### 7.2.1 IPv6-Adressierung

Bei IPv6 gibt es Unicast-, Multicast- und Anycastadressen. Eine Unicastadresse bezeichnet einen einzelnen Teilnehmer eines Netzwerkes (PC, Drucker, Router, etc.) während über Multicast- und Anycastadressen eine Gruppe von Teilnehmern erreicht werden kann. Der Unterschied zwischen Multicast und Anycast besteht darin, das ersteres alle Geräte einer Gruppe anspricht und zweiteres lediglich ein Gerät der Gruppe. Als Quelle von Nachrichten sind nur Unicast-Adressen zulässig.

Da die Schreibweise der IPv4-Adressen bei IPv6 zu viermal so langen Zahlensträngen wie bisher geführt hätte, hat man sich auf eine neue Schreibweise geeinigt. Die Adresse besteht aus 8 Blöcken zu je 4 Hexadezimalziffern[1], wobei führende Nullen in einem Block weggelassen werden dürfen. Neben weiterer Abkürzungsmöglichkeiten (siehe [1, S. 45]) gibt es zwei besondere Unicast-Adressen: „0:0:0:0:0:0:0:1“ entspricht localhost (127.0.0.1) und „0:0:0:0:0:0:0:0“ bedeutet „unspezifizierte IP-Adresse“.

Die IPv6-Adressen können durch Festlegen von Richtlinen für die ersten n Bits strukturiert werden. Grob umschrieben wird hier die Provider-/Servicehierarchie direkt in den Adressen umgesetzt um so das Routing zu erleichtern. So bekommen zum Beispiel die großen Provider einige Bits im vorderen Adressbereich zugewiesen. Der hintere Teil der Adresse enthält bei Unicast Adressen eine Interface-ID und bei Multicast-/Anycast-Adressen eine Gruppenkennung. Die 64 Bit lange Interface-ID wird entweder über eine Abbildung der MAC-Adresse oder der EUI-64-Adresse gewonnen. Dies ist bezüglich der Sicherheit fragwürdig. Genaueres dazu im Abschnitt 7.4.5

### 7.2.2 IPv6-Header

In IPv6 gibt es verschiedene Header, welche ein Paket spezifizieren. Es gibt einen Basis-Header (siehe Abb. 7.1) mit dem jedes IPv6-Paket beginnt und an den bei Bedarf per Verkettung Zusatzheader, wie zum Beispiel der „Authentication Header (AH)“ oder der „Encapsulation Security Payload Header (ESP)“ angehängt werden können. Dazu wird im Feld „Next Header“ der auf den aktuellen folgende Header eingetragen, somit findet man dieses Feld auch in jedem weiteren Header außer dem letzten Header vor der Nutzlast. Die Reihenfolge der Zusatz-Header ist zur Vereinfachung der Auswertung festgelegt.

[1] Beispiel: 4AF6:0854:BF45:78AB:10FF:5B8C:A54D:0FCD

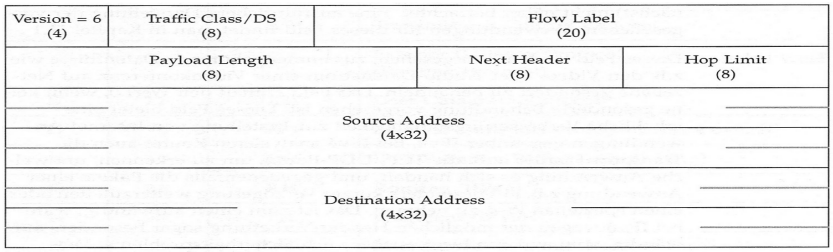


Abbildung 7.1: IPv6 - Basis-Header (aus [1, S. 19])

## 7.3 Sicherheit in Netzen

Zu den Zeiten als Netzwerke noch keine große Bedeutung hatten und auch das Internet noch nicht so populär war, gab es kaum Beweggründe um Netzkommunikation zu unterbinden, anzugreifen oder mitzuhören. Heutige Anwendungen wie z.B. Home-Banking, zunehmender E-Mail-Verkehr, e-commerce etc. bieten jedoch zunehmend Motivation für solche Attacken.

### 7.3.1 Bedrohungen

Die Bedrohungen für Kommunikation über ein Netzwerk lassen sich wie folgt gliedern:

- Aktiv
  - **Unterbrechen** der Verbindung als Angriff auf die Verfügbarkeit des Netzes
  - **Modifizieren** oder **Erzeugen** des Datenstroms unter falscher Identität als Angriff auf die Integrität und/oder Authentizität der Daten.
- Passiv
  - **Mithören/-lesen** des Datenstroms und somit Angriff auf die Vertraulichkeit der Daten

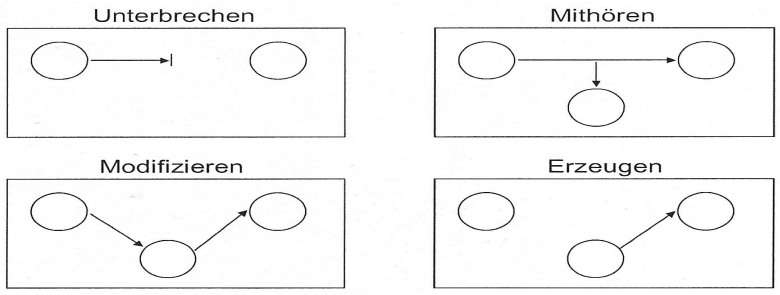


Abbildung 7.2: Bedrohungsarten in Netzwerken (aus [3, S. 3])

### 7.3.2 Gegenmaßnahmen

Für genaue Informationen bezüglich der Gegenmaßnahmen und Details zu den einzelnen Mechanismen, welche über diese Arbeit hinausgehen wird verwiesen auf Kapitel 1 dieses Seminars. Hier werden für die weitere Verwendung lediglich die Grundlagen erläutert.

Um den aktiven Bedrohungen entgegenzuwirken, sollten sich die Geräte in einem Netzwerk **authentifizieren** können, damit der jeweilige Kommunikationspartner sichergehen kann, dass er auch mit dem richtigen Gerät oder der richtigen Person in Verbindung steht. Zum Beispiel fügt der Absender hierzu mit Hilfe einer „Einweg-Funkion“ und einem Schlüssel sogenannte Authentifikationsdaten zum Datenstrom hinzu. Der Empfänger kann mit gleicher Funktion und gleichem Schlüssel diese Daten auch berechnen und eine Gleichheit authentifiziert das Gegenüber.

Die **Verschlüsselung** von Daten wirkt sowohl den aktiven als auch den passiven Bedrohungen entgegen, da sie die Daten für Dritte unbrauchbar macht. Es gibt symmetrische und auch asymmetrische Verschlüsselungsverfahren sowie zahlreiche Algorithmen, die beide kombinieren.

Ebenfalls ist es wichtig die **Datenintegrität**, zum Beispiel mit Hilfe von Prüfsummen, zu kontrollieren

Die beiden bereits erwähnten Header AH und ESP und die dahinterstehenden Mechanismen implementieren diese Maßnahmen in IPv6 und werden im folgenden Abschnitt genauer betrachtet.

## 7.4 Sicherheitsmechanismen bei IPv6 / IPSec

Der Bereich Sicherheit ist in IPv6 im Arbeitspaket IPSec zusammengefasst. Dieser Abschnitt behandelt im Detail die in IPv6 zur Verfügung stehenden Sicherheitsmechanismen.

Auf IP-Ebene gibt es zwei Arten von sicheren Kommunikationsbeziehungen (siehe Abb. 7.3). Zum einen den sogenannten *Transportmodus* („Normalfall“), welcher die sichere Verbindung zwischen zwei Endknoten darstellt und zum anderen den *Tunnelmodus*, der zwei private Netze über eine sichere Verbindung zweier Gateways zu einem virtuellen privaten Netz (VPN) zusammenschließt, wobei hier die Verbindungen innerhalb der privaten Netze weiter ungeschützt sind. Daher können beide Beziehungsarten auch kombiniert werden.

Technisch bedeutet der Tunnelmodus, dass beim Versenden von Paketen von einem Rechner aus Teilnetz 1 an einen Rechner in Teilnetz 2 vor das komplett fertige IP-Paket (mit allen Headern) von den Gateways nochmal eine Header-Kette gesetzt wird, welche die Informationen für die Kommunikation zwischen den beiden Gateways enthält.

Es ist klar, dass der Tunnelmodus mehr Overhead für die Steuerdaten benötigt, also der Nutzdatenanteil in den Paketen sinkt. Vorteil ist aber, dass der innere IP-Header voll durch den Sicherheitsmechanismus abgedeckt ist, was z.B. zur Folge hat, dass auch Quell- und Ziel-IP verschlüsselt sind (siehe Abb. 7.7). Der Unterschied der beiden Modi wird in den Abbildungen 7.5 und 7.7 deutlich.

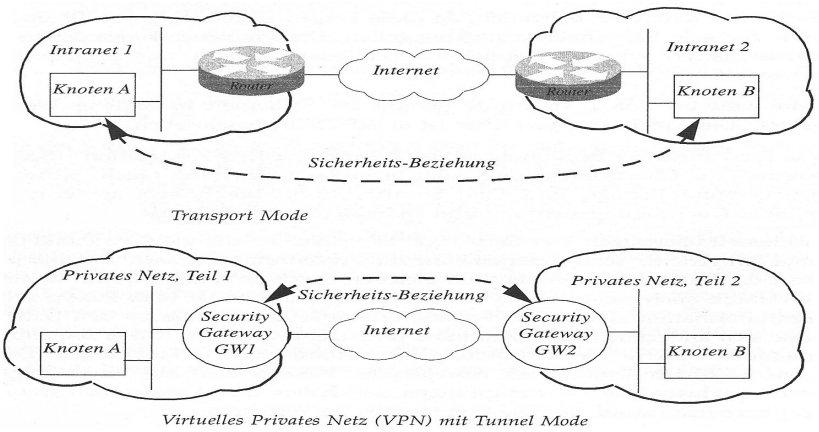


Abbildung 7.3: Kommunikationsbeziehungen (aus [1, S. 172/173])

### 7.4.1 Authentifikation und Integritätskontrolle

Für die Berechnung der angesprochenen Authentifizierungsdaten gibt es in IPv6 mindestens[2] die Verfahren „HMAC-MD5“ und „HMAC-SHA-1“. Beide sind Hash-Verfahren und erstellen aus den Daten und einem geheimen Schlüssel, mit Hilfe einer „one-way-hash“-Funktion, einen Wert, welcher dann zur Authentifikation und Integritätskontrolle dient. Einwegfunktion bedeutet, dass man mit diesem Algorithmus aus den Daten und dem Schlüssel den Hash-Wert zur Authentifikation bestimmen kann, die Rückrichtung aber nicht funktioniert.

MD5 arbeitet mit Schlüsseln ab 16 Bytes Länge, SHA-1 ab 20 Bytes. Nach oben hin ist der Länge keine Grenze gesetzt, jedoch werden Schlüssel mit einer Länge von mehr als 64 Bytes mit der Hash-Funktion auf 64 gekürzt und dann als modifizierter Schlüssel für die Berechnung genutzt. Somit werden in der Praxis meist Schlüssel mit Längen bis 64 Bytes genutzt. Folgender „Code“ soll einen kleinen Überblick über die Funktionsweise der Algorithmen geben:

```
ABKÜRZUNGEN:
  S    = Schlüssel
  N    = Nachricht
  hash = SHA-1- oder MD5-Hashfunktion
  ++   = Konkatenation

ABLAUF:
  S'   = S ++ 0..0                  // Auffüllen des Schlüssels mit Nullen
                                    // auf die nächste 64-Byte-Grenze
  Si   = S' XOR ipad                // ipad = 36 (Hex)
  So   = S' XOR opad                // opad = 5C (Hex)
  D    = hash(So ++ hash(Si ++ N))  // D = Ergebnis (=> berechneter Rest)
```

[2]die beiden beschriebenen Algorithmen sind in jeder IPSec-Implementierung vorhanden, es können jedoch weitere hinzugefügt werden

Beide Verfahren nutzen Hash-Funktionen, die im Wesentlichen einfache logische Operationen auf den Bytes ausführen und dann einen Restwert[3] als Ergebnis liefern. Dadurch sind die Berechnungen nicht zu aufwendig.

Die errechneten Authentifizierungsdaten müssen nun in den Datenstrom eingebunden werden. Dazu gibt es den „Authentication Header (AH)“ (siehe Abbildung 7.4).

Das *Next Header*-Feld ist das bereits vom Basis-Header bekannte. Die *Reserve* hat momentan noch den Standardwert 0 und dient der späteren Erweiterbarkeit des AH. *SPI* ist ein Pointer auf einen Eintrag in der „Security Association Database“ (siehe 7.4.3).

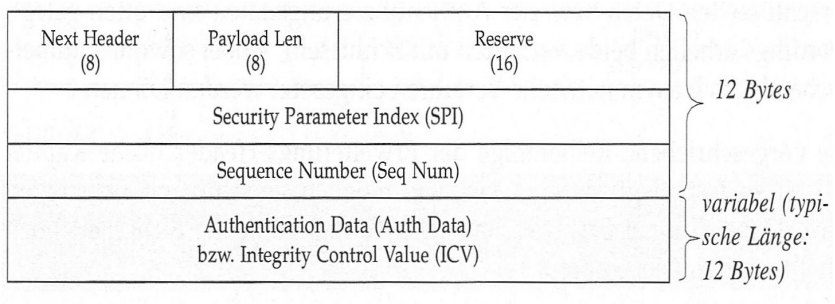


Abbildung 7.4: Authentication Header (aus [1, S. 182])

Die *Sequence Number* ist eine Zahl, welche beim Aufbau einer sicheren Verbindung mit 0 initialisiert wird und mit jedem Paket um 1 ansteigt. Bei Überlauf muss die sichere Verbindung neu aufgebaut werden. Diese Nummer verhindert sogenannte „Replay-Attacken“, bei denen Pakete mit identischen (Authentifizierungs-)Daten wiederholt werden würden.

Die *Payload Len* ist die Länge des gesamten AH als Vielfaches von 32 Bit.

Das Feld *Auth Data / ICV* enthält die ermittelten Authentifikationsdaten. Diese werden aus den Nutzdaten, allen Feldern des AH und den ihm nachfolgenden Headern sowie allen während des Transports nicht veränderbaren Feldern des Basis-Headers berechnet. Auch die Felder des Basis-Headers, die zwar während des Transports verändert werden, jedoch im Ziel vorhersagbar sind (z.B. Destination Adress) werden mit in die Berechnung eingebunden.

Authentifikation kann sowohl im Tunnelmodus, als auch im Transportmodus genutzt werden. Die entsprechend entstehenden Header-Ketten werden in der folgenden Abbildung (7.5) dargestellt.

Dort ist durch die dunkelgrauen Felder gut zu sehen, welche Daten tatsächlich authentifiziert werden. Ebenso wird der bereits erwähnte Unterschied zwischen Tunnel- und Transportmodus klar.

[3]MD5: 128 Bit ; SHA-1: 160 Bit

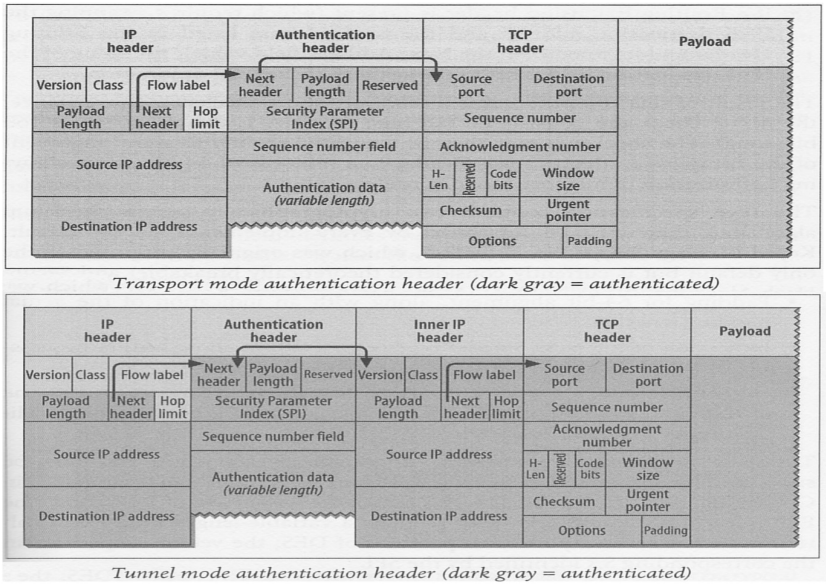


Abbildung 7.5: Header-Ketten bei Authentifikation (aus [2, S. 94/95])

Die Authentifikation soll sicherstellen, dass man Daten nur von den richtigen Geräten/ Personen empfängt, die Daten selbst jedoch sind völlig ungeschützt. Um das zu ändern müssen sie verschlüsselt werden.

### 7.4.2 Verschlüsselung

Bei Verschlüsselungsarten unterscheidet man zwischen symmetrisch und asymmetrisch. Beide Verfahren haben verschiedene Vor- und Nachteile. Ersteres benötigt einen geringeren Rechenaufwand als ein asymmetrisches Verfahren, der Nachteil jedoch ist, dass ein Schlüssel vorher ausgetauscht werden muss, während bei asymmetrischen Verfahren zur Verschlüsselung ein öffentlich bekannter Schlüssel verwendet wird und zur Entschlüsselung ein geheimer, nur dem Empfänger bekannter Schlüssel.

Beide Verfahren können auch kombiniert werden. Der Schlüsselaustausch für die symmetrischer Verschlüsselung der Nutzdaten kann über asymmetrische Verschlüsselung geschehen. Genaueres dazu wird in 7.4.4 erläutert.

In IPsec wird für die Verschlüsselung der „Encapsulation Security Payload Header (ESP)“ genutzt. Dieser ist offen für verschiedene Verfahren, in jeder IPsec-Implementierung muss jedoch ein Verschlüsselungsstandard enthalten sein, der „Data Encryption Standard (DES)“. Hierzu ein Auszug aus [9]:

*„DES funktioniert als Blockchiffre, das heißt jeder Block wird unter Verwendung des Schlüssels einzeln chiffriert, wobei die Daten in 16 Iterationen beziehungsweise Runden von Substitutionen und Transpositionen (Permutation) nach dem Schema von Feistel „verwürfelt“ werden. Die Blockgröße beträgt 64 Bits, das heißt ein 64-Bit-Block Klartext wird in einen 64-Bit-Block Chiffretext transformiert. Auch der Schlüssel, der diese Transformation kontrolliert, besitzt 64 Bits.*

*Jedoch stehen dem Benutzer von diesen 64 Bits nur 56 Bits zur Verfügung; die übrigen 8 Bits (jeweils ein Bit aus jedem Byte) werden zum Paritäts-Check benötigt. Die wirkliche Schlüssellänge beträgt daher nur 56 Bits. Die Entschlüsselung wird mit dem gleichen Algorithmus durchgeführt wobei die einzelnen Rundenschlüssel in umgekehrter Reihenfolge verwendet werden. [...] Weil die Schlüssellänge nur 56 bit beträgt, konnte DES bereits durch Brute Force-Angriffe gebrochen werden, indem systematisch alle Schlüssel getestet wurden. [...] Viele frühere DES-Nutzer benutzen jetzt Triple-DES auch (3DES) genannt, ein Verfahren, das von einem der DES-Mitentwickler beschrieben und analysiert wurde. Dabei wird jeder Datenblock mit einem Schlüssel chiffriert, mit einem anderen dechiffriert und wieder mit dem ersten chiffriert. Der gesamte Schlüsselraum hat damit eine Größe von 2112.“*

Mit Hilfe des ESP können die verschlüsselten Daten optional auch noch authentifiziert werden. Jedoch bezieht sich diese Sicherheit ausschließlich auf die Daten und nicht auf die Absender- oder Emfängerangaben. Um dies zusätzlich zu erreichen muss der ESP in Verbindung mit dem AH genutzt werden.

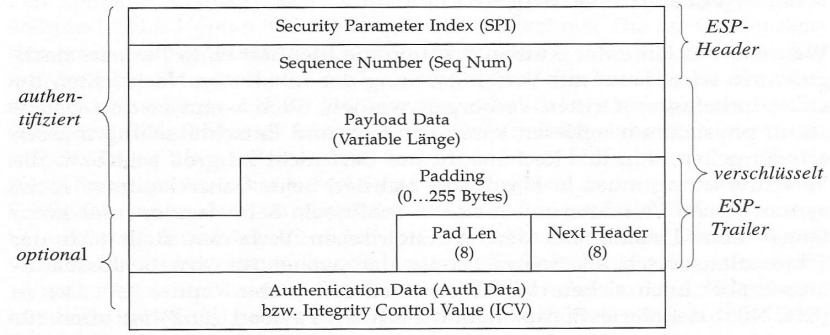


Abbildung 7.6: ESP-Header/-Trailer (aus [1, S. 186])

Im Gegensatz zu den bisher erwähnten Headern gehört zum ESP-Header auch noch ein ESP-Trailer. Im Header (siehe Abb. 7.6) befinden sich wieder die Felder *Seq Num* und *SPI* mit der gleichen Bedeutung wie im AH.

Darauf folgen dann die verschlüsselten Nutzdaten (*Payload Data*). Diese können mit beliebigen Daten (*Padding*) aufgefüllt werden, um z.B. eine bestimmte Länge zu bekommen und auch um die wirkliche Länge der Nutzdaten für potenzielle Angreifer zu verschleiern.

Diese „Dummy-Daten“ gehören bereits zum ESP-Trailer, welcher fortgesetzt wird mit dem Feld *Pad Len*, in dem die Anzahl der „Padding-Bytes“ enthalten ist.

Es folgt das bekannte Feld *Next Header* und abschließend das optionale Feld für die Authentifikationsdaten zu den verschlüsselten Nutzdaten (*Auth Data / ICV*).

Beim ESP gilt, wie beim AH, dass man ihn sowohl für den Transportmodus als auch für den Tunnelmodus benutzen kann. Die Abbildung 7.7 verdeutlicht dies und zeigt auch, durch die dunkelgrauen Felder, welche Bereiche tatsächlich verschlüsselt werden. Auch hier ist der technische Unterschied der beiden Modi erkennbar.

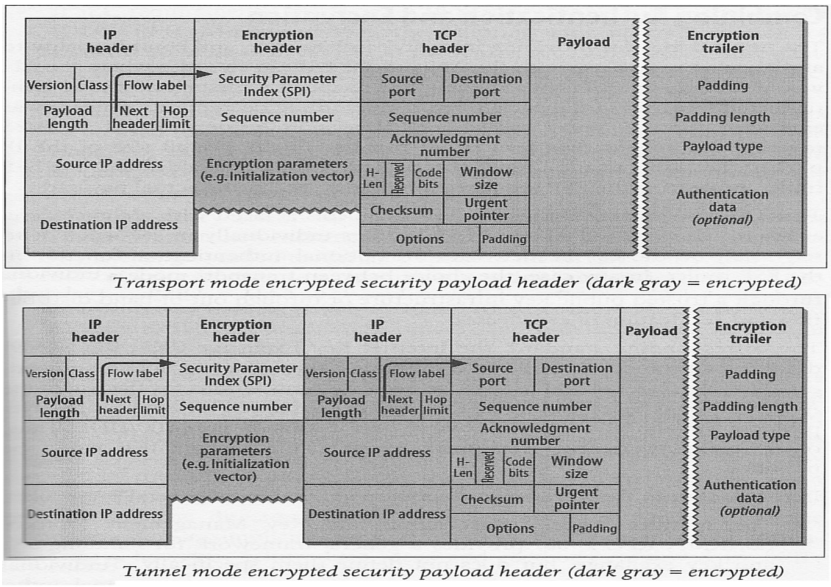


Abbildung 7.7: Header-Ketten bei Verschlüsselung (aus [2, S. 97])

### 7.4.3 Funktionsweise von IPsec

In diesem Abschnitt wird der prinzipielle Ablauf einer sicheren Kommunikation in IPsec erläutert.

#### Sicherheitsbeziehungen

Um in IPv6 überhaupt eine sichere Kommunikation mit Hilfe der bereits beschriebenen Mechanismen aufbauen zu können, müssen zwischen den Kommunikationspartnern zunächst sogenannte „Security Associations (SA)“ erstellt werden.

Eine SA ist in IPsec immer *unidirektional*, d.h. sie gilt nur in eine Kommunikationsrichtung. Des weiteren steht eine SA immer nur für einen Sicherheitsmechanismus, entweder Authentifikation oder ESP. Bei einer verschlüsselten und authentifizierten Verbindung in beide Richtungen werden also vier dieser Sicherheitsbeziehungen benötigt.

Eine SA ist ein Tripel, bestehend aus aus einem „Security Parameter Index (SPI)“, wie schon aus dem AH und dem ESP bekannt, aus einer IP-Zieladresse und einem Sicherheitsmechanismus (AH oder ESP). Die Zieladresse kann prinzipiell Unicast- oder Multicast-Adresse sein, aber zur Zeit wird lediglich der erste Fall von IPv6 unterstützt.

Alle Sicherheitsbeziehungen werden in einer „Security Association Database (SAD)“ gespeichert. Das Tripel selbst bildet das Schlüsselattribut dieser Datenbank, was auch der Grund für das SPI-Feld in den Sicherheits-Headern ist. Des weiteren werden für jede SA einige Komponenten, wie folgend aufgeführt, in der SAD abgespeichert.

- Die „Maximum Transmission Unit (MTU)“ für den Pfad der SA.
- Die Lebensdauer der SA
- IPsec-Modus (Tunnel- oder Transport-)
- Ein Zähler für die Sequenznummern sowie ein Flag für den Überlauf des selbigen.
- Alle Informationen bezüglich des verwendeten Verfahrens (Hash-Funktion, Verschlüsselungsverfahren, zusätzliche Daten, Initialwerte, etc.)
- Ein „Anti-Replay-Fenster“, also ein Zähler mit diversen Flags um Wiederholungen von alten (vor längerer Zeit gültigen) Paketen erkennen zu können.

Zusätzlich zur SAD gibt es in IPsec eine weitere Datenbank, die „Security Policy Database (SPD)“. In dieser werden die Sicherheitsstrategien für verschiedene IP-Pakete/ -Verkehrsarten gespeichert. Mit Hilfe von Selektoren werden diese unterschieden. Die Selektoren nehmen über verschiedene Angaben (Quell-Adresse, Ziel-Adresse, Benutzeridentifikation, Transportprotokoll, etc.) eine Klassifikation vor.

Somit können über die beiden Datenbanken und die jeweiligen Header immer die entsprechenden Mechanismen zum Aufbau und Durchführen einer sicheren Kommunikation angesteuert werden. Wobei hierbei eingehender und ausgehender Verkehr getrennt verarbeitet wird. Die Abläufe werden im Folgenden anhand der Abbildungen 7.8 und 7.9 erläutert.

### Ausgehender Verkehr

In der folgenden Abbildung wird der Ablauf für die Paketverarbeitung bei ausgehendem Verkehr dargestellt:

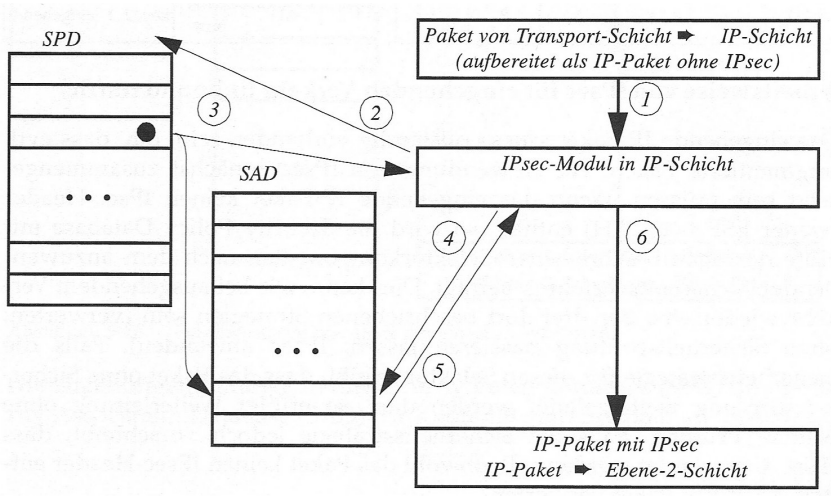


Abbildung 7.8: Ablauf bei ausgehendem Verkehr (aus [1, S. 177])

Der ausgehende Verkehr, der von der übergeordneten Schicht an die IP-Schicht übergeben wird (1), wird zunächst behandelt, wie es ohne IPsec auch geschehen würde. Anschließend wird aus der SPD bestimmt ob, und wenn ja welche Sicherheitsmechanismen für das aktuelle Paket benötigt werden (2). Die SPD kann folgende Entscheidungen für das Paket herbeiführen:

- das Paket wird verworfen
- es sind keine Maßnahmen erforderlich und es kann ungeschützt gesendet werden
- es müssen Sicherheitsmechanismen angewandt werden

Wenn noch keine SA vorhanden ist, wird ein Schlüssel-Austausch-Verfahren gestartet und eine neue SA eingerichtet (siehe Abschnitt 7.4.4). Wenn eine SA vorhanden ist, so gibt es einen Pointer in der SPD auf den entsprechenden Eintrag in der SAD. Dieser wird zurückgegeben (3) und damit das benötigte Verfahren und die SPI aus der SAD ermittelt (4)(5). Mit diesen Informationen werden die Sicherheitsmechanismen angewandt und die Header erstellt (6).

**Eingehender Verkehr**

Die nächste Abbildung zeigt den Ablauf für die Paketverarbeitung bei eingehendem Verkehr:

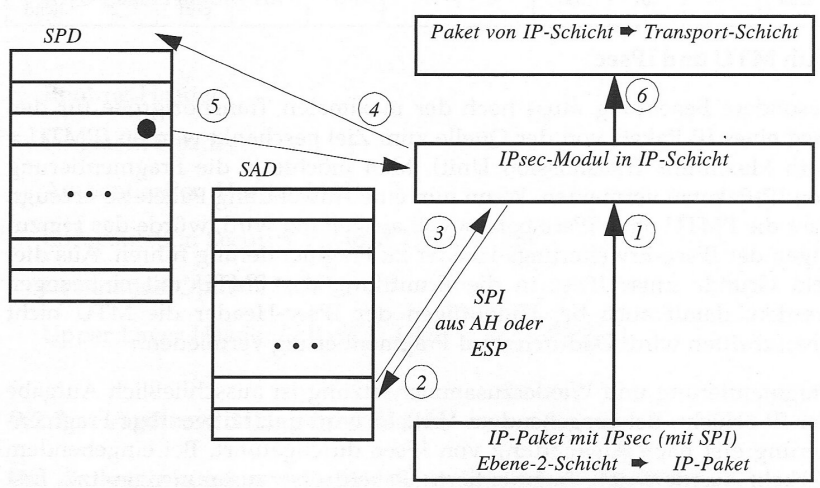


Abbildung 7.9: Ablauf bei eingehendem Verkehr (aus [1, S. 179])

Sobald ein IP-Paket vollständig angekommen ist, wird geprüft ob es einen IPsec-Header beinhaltet (1). Wenn dies der Fall ist, wird mit Hilfe der SPI des Headers in der SAD die entsprechende SA ausgewählt und mit dieser wird der IPsec-Header verabeitet (2)(3). Nachdem das Paket dann entschlüsselt und/oder authentifiziert wurde, wird mit Hilfe der (nun vorhandenen) Daten in der SPD nachgesehen, welches Verfahren hätte angewendet werden müssen (4)(5). Stimmt dieses Verfahren mit dem benutzten nicht überein, oder

stimmen gewisse Selektorkomponenten (Source, Destination, etc.) nicht mit den vorhandenen überein, so wird das Paket verworfen. Wenn nicht, wird es entschlüsselt weitergegeben an die nächsthöhere Schicht (6).

Sollte kein IPsec-Header zu finden sein, wird dennoch in der SPD nachgesehen, was für das Paket zu tun wäre. Hier können wieder die drei Fälle (Verwerfen, ohne Maßnahmen, mit Sicherheitsmaßnahmen) geantwortet werden. Sollte nun die SPD für ein Paket, welches ohne IPsec-Header übermittelt wurde, die Information beinhalten, dass dieses Paket gesichert sein müsste, so wird es verworfen.

Damit ist der eigentliche Ablauf der Sicherheitsmechanismen beschrieben. Doch ein elementarer Bestandteil einer sicheren Kommunikation, der Schlüselaustausch, wurde bisher noch nicht erläutert. Ohne diesen kann keine der bisher beschriebenen Sicherheitsmaßnahmen angewendet werden.

### 7.4.4 Schlüsselmanagement

IPv6 selbst legt keine Mechanismen für diese Vorgänge fest. Es gibt verschiedene Ansätze um den Schlüsselaustausch in IPv6 durchzuführen. Der am weitesten verbreitete ist es, einen Sicherheitskontext mittels „Internet Key Exchange (IKE)“ bereitzustellen.

IKE ist ein hybrides Protokoll[4] bestehend aus dem „Internet Security Association and Key Management Protocol (ISAKMP)“, dem „OAKLEY Key Determination Protocol“ und dem „Versatile Secure Key Exchange Mechanism for Internet (SKEME). Folgende Abbildung gibt einen Überblick über die Komponenten von IKE:

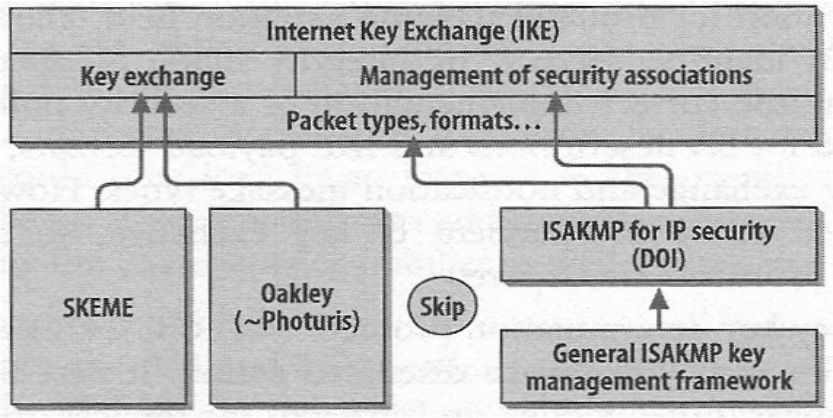


Abbildung 7.10: Überblick zu IKE (aus [2, S. 100])

IKE kann grob wie folgt beschrieben werden. Es handelt sich um ein Verhandlungsprotokoll zum Aushandeln einer Sicherheitsbeziehung, basierend auf Datenformaten, welche in ISAKMP definiert sind und Schlüsselaustauschmechanismen, welche auf OAKLEY und SKEME basieren.

[4]Ein hybrides Protokoll nutzt andere Protokolle oder Teile davon für seine eigene Funktionalität.

Der Vollständigkeit halber sei erwähnt, dass es noch zwei weitere Protokolle im „IKE-Bereich“ gibt, „Photuris“ und „SKIP“. Photuris entspricht im Wesentlichen dem „Main Mode“ von ISAKMP (siehe unten) und SKIP ist ein Schlüsselaustauschverfahren, welches wie OAKLEY auf dem „zero knowledge“-Prinzip beruht, jedoch werden hier die Schlüssel im Rahmen der Datenübertragung ausgetauscht. Genauere Informationen diesbezüglich sind zu finden in [3, S. 18-20]. Im folgenden wird der Ablauf von ISAKMP beschrieben.

IKE basiert auf UDP, nutzt standardmäßig den Port 500 und arbeitet in zwei Phasen um eine IPsec-SA zu erzeugen. In **Phase 1** wird per ISAKMP zwischen den Kommunikationspartnern eine sichere und authentifizierte Verbindung ausgehandelt. Zur Verschlüsselung und Authentifikation werden hierbei bekannte Verfahren (z.B. DES und MD5) genutzt. In dieser Phase werden auch Details zur IPsec-SA (Lebensdauer, zu benutzende Sicherheitsmechanismen, etc.) ausgehandelt. Das Ergebnis der ersten Phase wird auch ISAKMP-SA genannt. Im Gegensatz zur IPsec-SA ist diese bidirektional. Es gibt für diese Phase zwei Modi nach denen Verfahren werden kann. Der *Main-Mode* besteht aus drei Phasen, dem „Aushandeln der SA“, „Schlüsselaustausch“ und dem „Signieren“. Hierzu werden 6 Nachrichten zwischen den Kommunikationspartnern ausgetauscht. Der *Aggressive Mode* benötigt für das gleiche Ergebnis nur den Austausch von 3 Nachrichten. Dies geschieht durch Zusammenfassen der Phasen und auf Kosten der Sicherheit, denn es werden die Identitäten der Kommunikationspartner nicht geschützt. Der Schlüsselaustausch geschieht im Wesentlichen nach dem „Diffie-Hellman“-Prinzip (siehe [1, S. 190]). Beispielhaft folgt hier der Ablauf in den sechs Nachrichten des Main Modes:

1. Initiator sendet einen oder mehrere Vorschläge mit Authentifizierungs- und Verschlüsselungsalgorithmen
2. Antwortender wählt einen Vorschlag aus und bestätigt
3. Initiator sendet öffentlichen Teil der Diffie-Hellmann-Schlüsselvereinbarung und einen zufälligen Wert (Nonce)
4. Antwortender schickt ebenfalls öffentlichen Teil der Diffie-Hellmann-Schlüsselvereinbarung und einen zufälligen Wert (Nonce)
5. Initiator berechnet Signatur und sendet diese mit seiner Identität an Antwortenden. Diese Daten werden mit einem symmetrischen Schlüssel verschlüsselt.
6. Antwortender schickt gleiche Daten von seiner Seite an den Initiator

In **Phase 2** wird der Quick-Mode definiert. Basis zum Nutzen dieses Modus ist eine ISAKMP-SA aus Phase 1. Innerhalb der zweiten Phase werden Anwendungsstrategien für die in den vorangegangenen Abschnitten beschriebenen Sicherheitsmechanismen (ESP, AH) ausgehandelt und Schlüssel hierfür erzeugt und ausgetauscht. Diese Vorgänge benötigen dank der ISAKMP-SA nur noch drei Nachrichten.

### 7.4.5 Adressgeheimhaltung

Wie in Abschnitt 7.2.1 beschrieben, werden die IP-Adressen, bei Autokonfiguration, mit Hilfe hardwarespezifischer Informationen gebildet. Dieser Aspekt wird hinsichtlich der Bereiche „Privatsphäre“ und „Datenschutz“ bezüglich IPv6-Sicherheit auch diskutiert. Unter Umständen ist es dadurch möglich über die IP eines Nutzers Rückschlüsse auf seine Hardware und seinen „Bewegungsprofil“ zu ziehen. Um das zu verhindern wurde die Adressgenerierung auf Zufallsbasis eingeführt.

Im Netz wird bekanntlich unterschieden zwischen „Client“ und „Server“. Ein Server benötigt um seine Dienste vernünftig anbieten zu können eine bekannte und feste Adresse. Clients jedoch benötigen lediglich eine eindeutige Adresse, welche aber nicht bekannt und fest sein muss.

Die Idee ist nun, einem Knoten für Serverdienste, wie bisher, eine feste und bekannte Adresse zu geben und für die Client-Funktionen eine reproduzierbare Pseudo-Zufalls-IP-Adresse zu erstellen. Somit kann dem oben beschriebenen Missbrauch vorgebeugt werden.

Die folgende Abbildung zeigt, wie eine solche Adresse generiert wird:

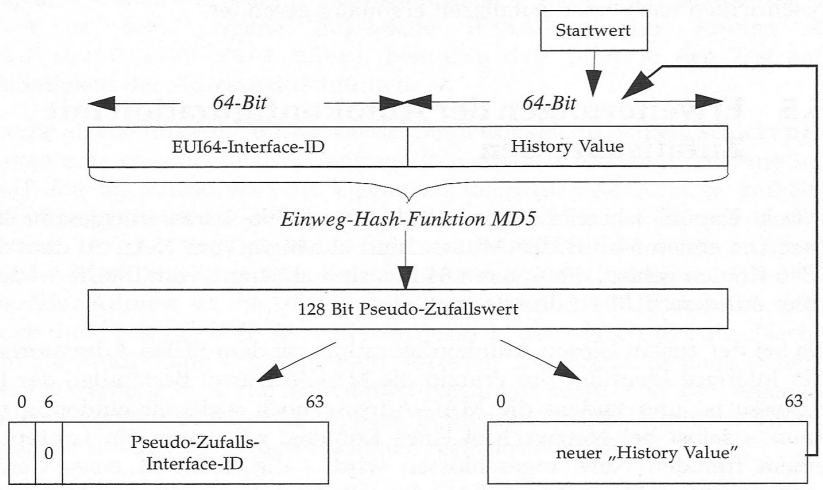


Abbildung 7.11: Generierung einer Zufalls-IP-Adresse(aus [1, S. 104])

Es muss ein 64 Bit langer Startwert übergeben werden (als erstes „History Value“) welcher kombiniert mit der EUI64-ID mittels einer Hash-Funktion in einen 128 Bit langen Zufallswert umgewandelt wird. Dieser dient zur ersten Hälfte als Zufalls-Interface-ID für die zu generierende Adresse und zur zweiten Hälfte als „History-Value“ für die nächste Generation.

Im unwahrscheinlichen Fall einer doppelt vorkommenden IP-Adresse wird eine neue generiert. Genaueres dazu findet man in [1, S. 103-105]

## 7.5 Nutzen von IPv6 für die Sicherheit

In diesem Abschnitt soll erklärt werden, was IPv6 konkret für die Sicherheit in Netzwerken bringt. Hierzu mache man sich klar, dass IPsec Sicherheit in der Vermittlungsschicht, also bei der Paketübermittlung zwischen zwei Geräten schafft.

Es gibt Diskussionen darüber, in welcher Schicht Sicherheitsmechanismen implementiert werden sollen und einige meinen, dass man dies den Anwendungen gänzlich überlassen sollte.

Der Hauptvorteil von IPsec liegt meiner Meinung nach darin, dass die Sicherheit einer so niedrigen Schicht implementiert wird und damit ein Standard für alle Nutzer mit einer großen Unabhängigkeit von Anwendungsprogrammen, Betriebssystemen usw. geschaffen wird. Sicherheitsmechanismen auf einer höheren Ebene sollen Lücken in der Sicherheit unterer Schichten ausgleichen und mit IPsec wurde eine große Lücke, die Unsicherheit der IP-Pakete, geschlossen. Außerdem ist IPsec transparent für die Anwendungen, d.h.:

- Auch Anwendungen die keine eigenen Schutzmechanismen haben können geschützt werden.
- Es wird ein Sicherheitskontext aufgebaut und alle Anwendungsdaten sind geschützt (z.B. bei der Kopplung von Netzen)

IPv4 ist immer noch das am meisten verbreitete Protokoll und IPsec wurde auch für IPv4 nutzbar gemacht. Aus dem Blickwinkel der Sicherheit könnte man nun dazu tendieren zu fragen: „Warum noch IPv6 ?“

### 7.5.1 IPv6 vs IPv4

IPsec kann in IPv4 zwar auch genutzt werden, jedoch kann man sich nie darauf verlassen, dass ein Kommunikationspartner die Sicherheitsmechanismen auch wirklich unterstützt, denn IPsec ist aus Kompatibilitätsgründen nicht verpflichtend in IPv4 übernommen worden sondern nur optional.

IPv6 stellt immer die Grundfunktionen von IPsec zur Verfügung, so dass ein Standard, eine gemeinsame sichere Sprache, vorhanden ist.

Aus Gründen der Adressknappheit wurden in IPv4 diverse Verfahren eingesetzt um schonender mit der Menge von Adressen umzugehen. So wurden zum Beispiel Netze mit dem Prefix 192.168 zur privaten Nutzung freigegeben. Diese Netze können dann mit Hilfe von NAT (Network Adress Translation) über eine öffentliche IP (meist vom Provider zugeordnet) mit dem Internet kommunizieren. Das Problem von NAT ist, dass hier kein weiterer Header angehängt wird, wie beim Tunnelmodus, sondern die IP-Adressen im Header geändert werden. Dieser Mechanismus bricht die Arbeitsweise von IPsec. Das hat zur Folge, dass weitere Mechanismen zur effektiven Nutzung von IPsec (wie zum Beispiel eine „NAT-Erkennung“) gebraucht werden.

Dies ist durch den ausreichen großen Adressraum von IPv6 nicht nötig, da man auf Mechanismen wie NAT hier verzichten kann. Noch dazu sei gesagt, das IPsec während der Entwicklung von IPv6 entstand, und somit ohne das neue Internetprotokoll evtl. noch gar nicht in dieser Form existieren würde.

## 7.6 Abschließende Bemerkungen

IPsec trägt zur Sicherheit des Internets und anderer Netzwerke bei und eignet sich bestens für den Aufbau von VPNs. Die Authentifikation von Online-Inhalten (Applets, ActiveX, etc.) wird erheblich erleichtert und auch der Austausch vertraulicher Informationen wird effektiv unterstützt. Darauf aufbauend aber Aussagen zu treffen wie „E-Commerce wird durch IPv6 sicherer“ oder gar „IPv6 schützt den Rechner“ ist problematisch, denn die Sicherheit von Rechnern, E-Commerce und ähnlichen Anwendungen hängt von viel mehr Faktoren ab als nur dem sicheren Austausch von IP-Paketen (siehe Kapitel 1 dieses Seminars).

Sicherheit auf IP-Ebene bringt Vorteile mit sich:

- Wie bereits erwähnt, ist eine große Unabhängigkeit von Anwendungssoftware, Betriebssystemen etc. gegeben
- Die implementierten Verfahren können sowohl im End-to-End- als auch im Netz-Bereich eingesetzt werden, was einen gewissen Grad an Flexibilität bedeutet.

Selbstverständlich ist IPsec noch weit von Perfektion entfernt. Es gibt einiges an negativer Kritik bezüglich der Arbeitsweise von IPsec. Hauptansatzpunkt der Kritiker ist die Komplexität des Sicherheitspakets.

- Die Kombinationsmöglichkeiten von AH und ESP komplizieren die Abläufe. Es kann verschlüsselt werden, ohne Authentifikation, authentifiziert werden ohne Verschlüsselung oder beides in Kombination eingesetzt werden.
- IKE mit seinen diversen Modi und Ablaufarten trägt zur Komplexität der Sicherheitsmechanismen unter IPv6 bei.

Trotz aller Kritik trifft folgende Aussage von „Niels Ferguson“ und „Bruce Schneier“[5] die Bedeutung von IPsec ziemlich genau:

*„Even though the protocol is a disappointment–our primary complaint is with its complexity–it is the best IP security protocol available at the moment.“*

[5]Experten auf dem Gebiet der Kryptografie, welche IPsec in „A Cryptographic Evaluation of IPsec“ unter die Lupe genommen haben

# Literaturverzeichnis


[1] Wiese, Herbert: *Das neue Internetprotokoll IPv6*
2002 Carl Hanser Verlag; ISBN 3-446-21685-5

[2] Hagen, Silvia: *IPv6 Essentials*
2002 O'Reilly Verlag; ISBN 0-596-00125-8

[3] Dr. Lubich, Hannes P.: *IP Next Generation - Sicherheitsdienste in IPv6*
www.tik.ee.ethz.ch/kurse/ipng/Folien-PDF/005-IPv6_Security.pdf

[4] Fritsche, Wolfgang: *Tutorial: IPv6 mobility and security*
www.seinit.org/documents/IPv6%20mobility%20and%20security.pdf

[5] Dittler, Hans-Peter: *IPv6 - Tutorial - Development of IPv6 - Exploring IPv6*
linda.ipv6.berkom.de/summit/01-02_Hans-Peter.Dittler_Ipv6-tut1.pdf

[6] Thomas, Wolfgang: *IPsec Architektur und Protokolle, Internet Key Exchange (IKE)*
www.net.informatik.tu-muenchen.de/teaching/WS02/security/securityUeb/07ausarbeit.pdf

[7] IPsec ist spezifiziert in folgenden RFC's einzusehen auf *www.ietf.org/rfc.html*
RFC 2401 (Sicherheitsarchitektur für das Internetprotokoll)
RFC 2402 (Authentication Header)
RFC 2406 (Encapsulation Security Payload)
RFC 2407 (IPsec Domain of Interpretation)
RFC 2408 (ISAKMP)
RFC 2409 (IKE)

[8] Beitrag über IPsec von *www.ipv6-net.org/themen/uebe/page12.php#7_1*

[9] Diverse Begriffsdefinitionen (IPsec, IPv6, ESP, IKE, ...)
nachgeschlagen bei *www.wikipedia.org*

# Kapitel 8

# Digital Rights Management

*Christopher Mader*

*In dieser Arbeit sollen in erster Linie Methoden, Grundlagen, Techniken und Technologien für DRM-Systeme erläutert werden. Neben dem technischen Bereich soll aber auch Aspekte im Umgang des Menschen mit dieser Technologie eingegangen werden. Es werden zum Beispiel gesetzliche Grundlagen knapp erläutert, aber auch Bedenken von vehementen DRM-Gegnern angesprochen.*
*Die Hauptabschnitte Technik und Sicherheit sind mit bedacht so allgemein gehalten, dass ihr Inhalt trotz der Vielzahl an DRM-Systemen allgemeingültig ist. Im Gegensatz zu den meisten anderen Quellen zu DRM legt der Autor dabei auf eine möglichst objektive Darstellung der Sachverhalte Wert.*
*Um dennoch eine grobe Vorstellung der unterschiedlichen Auffassungen zu DRM zu vermitteln sind die Kapitel zur Motivation und den Befürchtungen jeweils recht einseitig gehalten. Die Beschreibung der rechtlichen Grundlagen ist als kurze Einführung gedacht und soll einen Startpunkt für die weitere Recherche in diesem Bereich bieten.*

## 8.1 Motivation

DRM ist in letzter Zeit in aller Munde und wird in der öffentlichkeit kontrovers diskutiert. Hierzu möge der geneigte Leser bei weiterem Interesse einmal nach 'DRM' oder 'Digital Rights Management' googlen. Diese Suche bringt in etwa 7.000.000 Einträge hervor[1]. Tilo Stadelmann zieht in seiner Veröffentlichung über DRM [4] als Bedeutungsbarometer unter Anderem die etlichen Berichte der Computerzeitschrift C't in den vergangenen Jahren heran. Ein weiteres Indiz für die zukünftige Bedeutung von DRM-Techniken sind die vielen Firmen, die selber DRM-Systeme anbieten oder entwickeln beziehungsweise sich solchen Anbietern angeschlossen haben. Zum Beispiel besteht das Open-eBook-Forum derzeit aus einer großen Anzahl von Mitgliedern. Die Bedeutung von DRM ist im Wesentlichen auf zwei Kernbereiche zurückzuführen. Zum Einen der Schutz vor Raubkopien und zum Anderen die Hoffnung auf einen neuen Absatzmarkt. Aus einer Erhebung der Business Software Alliance geht hervor, dass im Jahr 2003 Software im Wert von 80 Milliarden Dollar installiert wurde. Allerdings wiesen nur 64 Prozent rechtmäßige Lizenzen auf. Daraus ergibt sich eine Summe von 29 Milliarden Dollar an entgangenen Einnahmen durch Raubkopien. Gleichzeitig prognostiziert der Verband eine Steigerung des Kopiervolumens auf 40 Milliarden Dollar pro Jahr im Laufe der nächsten fünf Jahre. Daraus geht hervor, wie wichtig ein effektiver Schutz digitaler Inhalte für die Wirtschaft geworden ist. Auch der volkswirtschaftliche Schaden ist nicht zu vernachlässigen. InterTrust Technologies, einer der großen Anbieter von DRM-Systemen, verspricht mit seinem DRM die nahezu unbegrenzten Möglichkeiten des Internet zu entfesseln:
*'Digital Rights Management (DRM) is an umbrella-term for new business trust assurance processes designed to unleash the tremendous capabilities of the internet. DRM technology provides tools to enable these new processes.'[2]*
Renato Ianella spricht in 'DRM Architectures' [1] im Zusammenhang mit DRM nicht nur von einem Management auf digitalen Inhalten, sondern vom digitalen Management jeglicher Urheberrechte. Diese feine Unterscheidung wird allerdings in dieser Arbeit nicht weiter verfolgt, da das Ziel die Vorstellung und Erklärung der grundlegenden Mechanismen und Techniken des DRM ist.

[1] stand vom 25.06.2005

## 8.2 Rechtliche Grundlagen

Digital Rights Management dient zum Schutz der Rechte der Urheber. Diese sind in Deutschland durch das Urheberrechtsgesetz (UrhG) geschützt. Dieses Gesetz wird häufig im Privaten wie in der Forschung leicht gedehnt um die Inhalte im Sinne des amerikanischen 'fair-use' zu nutzen. Darunter fallen in Deutschland zum Beispiel die Privatkopie, und insbesondere auch das Zitieren in wissenschaftlichen Arbeiten.
Nach einem Zitat aus einem Interview der Fachzeitschrift C't [9] mit der Bundesjustizministerin Renate Zypries und Ministerialdirektor Dr. Elmar Hucko kennt das Urheberrecht jedoch kein Recht auf die Privatkopie. 'Es gibt nur Schranken des Urheberrechts, das heißt, der Rechteinhaber muss Vervielfältigungen zum privaten Gebrauch dulden und bekommt im Gegenzug seinen Anteil an der Pauschalvergütung. Die Zulässigkeit der Privatkopie beruht auf einer staatlichen Lizenz nach dem Motto: Schützen, was man schützen kann, vergüten, was man nicht schützen kann.'
Nun sollen Content-Anbieter durch DRM in die Lage versetzt werden Inhalte beliebig zu schützen. Dies verhindert folgerichtig die Privatkopie, insbesondere aber auch das Zitieren im Sinne der Forschung.
'Ein anschauliches Beispiel für die bevorstehenden Konsequenzen der Gesetzgebung brachte auch Bernd Lutterbeck. Der Informatikprofessor an der TU Berlin hatte just in der Entscheidung des obersten US-Gerichtshofs zur Bestätigung der Verlängerung der Copyright-Fristen jenseits des Atlantiks um 20 Jahre in einem Artikel in der Londoner Times gelesen, demzufolge der US-Medienkonzern Time-Warner nach dem Aufkauf der Rechte des 1893 von zwei US-Amerikanischen Lehrern geschriebenen Geburtstagsständchens 'Happy Birthday' jährlich rund 2mio US-$ Lizenzeinkommen einfährt. Die Rechte gelten noch bis 2009. Lutterbeck wollte den aus sechs Sätzen bestehenden Bericht gerne als normale Kopie zur Pressekonferenz mitbringen. Das gestatte der Times-Verlag – allerdings bei 100 analogen Vervielfältigungen für stolze 50 britische Pfund. Würde er seine Arbeitszeit hinzurechnen, so Lutterbeck, käme der Spaßauf 387 Euro inklusive Steuern.' Offensichtlich würde eine solch exorbitante Preisgestaltung vielen Mitgliedern der akademischen Gemeinschaft eine Veröffentlichung erschweren.
'Der Gesetzgeber hätte private Vervielfältigungen schon vor 40 Jahren verboten, wenn er ein Verbot im Sinn gehabt hätte,' so Hucko, 'aber das funktioniert nicht: Das wäre ebenso effektiv gewesen wie ein Verbot des Nasebohrens.'
In Europa wurde mit der 'Directive of the European Parliament and of the Council on Measures and Procedures to ensure the Enforcement of Intellectual Property Rights' in Anlehnung an den US-Amerikanischen 'Millenium Digital Copyright Protection Act' eine Richtlinie zum Schutz des Kopierschutzes geschaffen. Diese Richtlinie wurde in Deutschland mit dem 'Gesetz zur Regelung des Urheberrechtes in der Informationsgesellschaft' umgesetzt. Im Kernabsatz §95a wird in diesem Gesetz das Umgehen eines Kopierschutzes zur gewerblichen Nutzung verboten. Damit hat Deutschland im Gegensatz zu den USA einen günstigen Kompromißzwischen den Begehren der Urheber und der Endnutzer geschlossen.
Verschiedene Zeitgenossen sehen in DRM eine Gefährdung der Datenschutzbestimmungen. 'Das Bundesverfassungsgericht hat im Volkszählungsurteil vom 15.12.1983 erstmals anerkannt, dass es ein Grundrecht auf informationelle Selbstbestimmung gibt:
*'Das Grundrecht gewährleistet insoweit die Befugnis des Einzelnen, grundsätzlich selbst*

*über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen. Einschränkungen dieses Rechts auf 'informationelle Selbstbestimmung' sind nur im überwiegenden Allgemeininteresse zulässig.'*
Es besteht demnach ein *'Schutz des Einzelnen gegen unbegrenzte Erhebung, Speicherung, Verwendung und Weitergabe seiner persönlichen Daten'. Das Grundrecht auf informationelle Selbstbestimmung wird als besondere Ausprägung des schon zuvor grundrechtlich geschützten allgemeinen Persönlichkeitsrechts angesehen. Wie dieses wird es verfassungsrechtlich aus Art. 2 Abs. 1 (sog. allgemeine Handlungsfreiheit) in Verbindung mit Art. 1 Abs. 1 GG (Menschenwürde-Garantie) hergeleitet.'* [3]
Gerade die in unter 'Technik' beschriebene Onlinevariante der Sicherung ermöglicht es dem Betreiber eines DRM-Systems, in schlecht nachvollziehbarer Weise Daten über den Nutzer zu erheben, die über das notwendige Minimum hinausgehen. Damit wäre sogar möglich, zum Zwecke der gezielten Werbung Interessenprofile zu erstellen.

## 8.3 Technik

Dieser Bereich beschäftigt sich mit technischen Fragen zu DRM. Zum einen der grundlegende Aufbau eines Gesamten Systems um einen groben Überblick zu bekommen und die vielen verschiedenen Aspekte des DRM aufzuzeigen. Zum anderen die eigentliche Beschreibung der Rechte und zuletzt die Sicherung des Inhaltes und des Systems.
Im Folgendem ist von Content die Rede als ein von dem DRM-System zu schützendem Inhalt. Dabei spielt es keine Rolle, ob damit Bilder, Musik, Video oder einfach nur Text gemeint ist. Content kann auch eine Zusammenstellung von verschiedenen anderen Inhalten sein.

### 8.3.1 Schemata

Im Folgenden werden zunächst zwei Schemata im Bereich des DRM erläutern. Das Vertriebsschema verschiedener Anbieter könnte jeweils als eigenes Schema aufgefasst werden, darauf möchte ich aber nicht näher eingehen, da es von Implementierung zu Implementierung unterschiedlich ist. Stattdessen werden an dieser Stelle die beiden Schemata aufzeigen, die Ianella [1] Functional Architecture und Information Architecture nennt.

#### Funktionelle Architektur

Das gesamte DRM-Framework um DR-fähige Systeme zu erstellen kann in drei Bereiche gegliedert werden:

- **Erstellung und Erfassung von Gütern geistigen Eigentums *(Intellectual Property (IP) Asset Creation and Capture)*:** Wie erstellt man einen Content so, dass er sich zum Vertrieb eignet. Dieser Bereich schließt die Zusicherung von Rechten bei Erstellung des Contents durch verschiedene Content Creators oder Vertreiber mit ein.
- **Management von Gütern geistigen Eigentums *(IP Asset Management)*:** Wie verwaltet man den Content und bereitet ihn zum Vertrieb vor. Das Vertriebssystem muss die deskripten Metadaten sowie die rechtebeschreibenden Daten verwalten, also beteiligte Parteien, Nutzungen, Zahlungen etc.
- **Nutzung von Gütern geistigen Eigentums *(IP Asset Usage)*:** Wie verwaltet man die Nutzung des Contents nach dem Vertrieb. Dies schließt Nutzungsbeschränkungen des vertriebenen Contents in spezifischen Nutzerumgebungen mit ein.

Die Functional Architecture beschreibt die Rollen und Handlungsbereiche einer Anzahl kooperierender und interoperierender Module der drei oben dargestellten Bereiche der Nutzung geistigen Eigentums. Diese sollen im Folgenden näher betrachtet werden:

Das Modul zur Erstellung und Erfassung von Gütern geistigen Eigentums umfasst:

- **Rechtevalidierung** - um sicherzustellen, dass die Erschaffung von Content aus bereits existierendem Content erlaubt ist.
- **Rechteerschaffung** - um dem neugeschaffenen Content Rechte zuweisen zu können, zum Beispiel um den Besitzer und Nutzungsbeschränkungen festzulegen.
- **Rechteverarbeitung** - um den Content eine Reihe von Verarbeitungsschritten durchlaufen zu lassen, bei denen die vergebenen Rechte und der Inhalt selbst überarbeitet werden können.

Das Modul zum Management von Gütern geistigen Eigentums umfasst:

- **Zugriffsfunktionen** - um den Zugriff auf (eventuell verteilte) Datenbanken mit Inhalten und zugehörigen Metadaten zu gewährleisten.
- **Vertriebsfunktionen** - um die Vergabe von Lizenzen an Parteien zu gewährleisten, die Rechte über den Content erworben haben. Dies schließt Zahlungen von Nutzern an die Rechteinhaber ein.

Das Modul zur Nutzung von Gütern geistigen Eigentums umfasst:

- **Rechteverwaltung** - zum Durchsetzen der mit den Inhalten verknüpften Rechte. Zum Beispiel könnte nur die Betrachtung des Inhaltes erlaubt sein, nicht aber die Änderung.
- **Nachverfolgung** - erlaubt es, sofern die Vertragsbestimmungen es erlauben, Nutzungsdaten zu erfassen und eine vertragsgemäße Nutzung sicherzustellen. Zum Beispiel könnte ein Nutzer nur die Lizenz erworben haben, ein Video zehnmal zu betrachten.

Zusammen bilden diese drei Module das Grundgerüst für ein DRM-System. Sie wurden hier nur oberflächlich beschrieben und müssen in der Praxis untereinander und mit bestehenden Geschäftsmodellen kooperieren und interoperieren.

Die Funktionelle Architektur kann nur eine Teillösung für die Herausforderungen eines DRM-Systems sein. Die Rechteverwaltung gewinnt sehr schnell an Umfang und Komplexität. Dies erfordert, dass DRM-Systeme die flexibelsten verfügbaren Informationsmodelle unterstützen müssen. Dies leistet die Information Architecture.

### Information Architecture

Es existieren drei grundlegende Größen in der DRM-Umgebung: der User, das Recht und der Inhalt an sich. Der Benutzer ist hier nicht nur auf den Endanwender beschränkt, sondern auch der Rechteinhaber und der Urheber sind User im Sinne der Definition. Da verschiedenste Relationen zwischen diesen Größen bestehen können, müssen Rechtedefinitionssprachen (Right Definition Languages - RDL) Beziehungen zwischen diesen

formulieren können. Diese drei Grö§en müssen jeweils noch weiter beschrieben werden. Nach Ionella sollten dabei bewährte Methoden verwendet werden, z.B. für den Benutzer die vCard.

Der Inhalt soll in verschiedene Ebenen unterteilt sein. Von der International Federation of Library Associations (IFLA) existiert ein Modell, das in verschiedene Schichten aufgeteilt ist. Aus Mangel an besseren Bezeichnungen im Deutschen werden hier die Originalbezeichnungen verwendet. Die oberste Schicht dabei ist das Work, also die tatsächliche geistige Leistung des Erschaffers, z.B. der Originaltext eines Buches. Hinzu kommt die Expression, also die Ausdrucksform. Dies kann beispielsweise derselbe Text sein, übersetzt in eine andere Sprache. Die Manifestation beschreibt die Art der Darreichung, also beispielsweise ob das Buch als gebundene oder Taschenbuchausgabe, als Hörbuch oder in digitaler Form existiert. Als letzte Schicht kommt die tatsächliche Instanz, bzw. das Item. Dies ist dann, um im Beispiel zu bleiben, ein Exemplar eines ins Deutsche übersetzten Textes in gebundener Form. Der große Vorteil dieser Beschreibung liegt darin, dass die Rechte der einzelnen Schichten unabhängig voneinander verwaltet werden können.

Neben der genauen Beschreibung der einzelnen Größen müssen diese auch eindeutig identifizierbar sein. Auch hierfür fordert Ionella öffentliche, bekannte und bewährte Verfahren. Zur Identifikation von Inhalten möchte ich den Leser an dieser Stelle auf Digital Object Identifier (DOI) verweisen. Dieses Konzept ist zur Zeit im Entstehen begriffen. Zur Rechtebeschreibung gibt es noch kein einheitliches Verfahren, womit wir zu den Right Definition Languages kommen.

### 8.3.2 Right Definition Languages

Right Definition Languages sollen zur Beschreibung unterschiedlichster Rechte dienen. Dabei bezieht sich ein Recht immer auf das Quadrupel User, Inhalt, Promise, Beschränkungen(Constraints). User beschreibt hierbei einen oder eine Gruppe von Benutzern, für die dieses Recht gilt. Mit Inhalt ist jedweder geartete Inhalt gemeint, auf den sich das Recht bezieht. Promise (Versprechen) sind die Rechte, die der User vom Rechteinhaber erhält, z.B. nur Abspielen oder aber Abspielen und eine gewisse Anzahl von Kopien erstellen. Beschränkungen bezeichnen die Bedingungen, unter denen dieses Recht zu gewährleisten ist. So wäre es zum Beispiel denkbar, dass ein Auszug aus einem Buch für den Kritiker nur bis zu einem bestimmten Zeitpunkt lesbar ist.

Eine unabdingbare Forderung ist die Maschinenlesbarkeit. Um dieses zu gewährleisten, setzen die meisten Produkte auf XML als plattformunabhängiges Format. Eine weitere Forderung ist die möglichst speicherplatzsparende Beschreibung dieser Elemente, da global gesehen ein gewaltiges Volumen an Metadaten erzeugt wird. Zwei Implementierungen hierzu sind XRML und ORDL, die im Abschnitt über Projekte etwas näher beschrieben sind. An dieser Stelle möchte sei DRI/DRD das Mittel eine Sprache näher zu erläutern. DRI/DRD sind die beiden Ausprägungsformen der Digital Rights Definition Language (DRDL). Diese wurde von der Internet Engineering TaskForce (IETF) als Beispiel für eine RDL vorgeschlagen. DRI steht hierbei für Digital Rights Instance, DRD für Digital Rights Description. Der Hintergedanke dieser Aufteilung ist das oben angedeutete Speicherplatzproblem. Eine DRI enthält nur die notwendigsten Informationen, die für

die personalisierte Speicherung eines Rechtes benötigt werden. Um die vollständigen Informationen für die Evaluierung zu erhalten, enthällt sie den eindeutigen Hashwert der zugehörigen DRD. Ein Beispiel für ein DRI eines Event-Tikets:

```
<DigitalRightInstance>
<Right ID='452FF96DEAF7E4379BB2E7C883C2CA53'>
<Issuer fingerprint='C883C2CAE4379BB2E753FFDEAF796452'/>
<IssueDate>1999/03/31T15:18:00</IssueDate>
<Owner ooms=http://ooms.trade.ietf.org/>
<Promise>Seat:A-203</Promise>
<Signature>
iQCVAwUANvoM8GACOp6X5M3tAQGplgP/a6/3l16a4tsfOcT2I+h6m
0Z1W7TkOH1OBIBa9GhHeAXy+e2HqccL9QTJlVnutU2qKTRnxCgx
+EDOxq8iJuQ=JkdKeYxH3OWos0CPThwkTDK5
</Signature>
</DigitalRightInstance>
```

Die Felder im einzelnen:
Right ID: Eindeutiger Bezeichner des DRD
Issuer fingerprint: Signatur des Herausgebers
IssueDate: das Ausgabedatum
Owner ooms: 'Online ownership management system', ein Verweis auf das zu verwendende IP Asset Management
Promise: Das eigentliche Recht das dem Besitzer dieser DRI versprochen wird (in diesem Fall eine Platznummer)
Signature: der Beweis der Authenzität der DRI

Das Digital Right (DRD) dazu:

```
<DigitalRightDefinition ID="452FF96DEAF7E4379BB2E7C883C2CA53Ó>
<Name>An event ticket</Name>
<CirculationConditions>
<Issuer HasRight="C883C2CAE4379BB2E753FFDEAF796452' />
<Collector HasRight='C883C2CAE4379BB2E753FFDEAF796452Ó />
</CirculationConditions>
<Validity times="1" end=Ò1999/03/31T14:18:00Ó/>
<View resource=Òhttp://trade.ietf.org/drti/icons/a001.gifÓ/>
<Promise resource=Òhttp://trade.ietf.org/drti/a001.htmlÓ/>
<SignatureProperties>
<CanonicalizationMethod Algorithm='http://www.w3.org/xml-c14n'/>
<SignatureMethod Algorithm='dsig:dsaWithSHA-1'/>
</SigtureProperties>
</DigitalRightDefinition>
```

Diese DRI berechtigt also zum einmaligen (Validity times='1') benutzen des Sitzplatzes A-203 zu einem gegebenen Zeitpunkt. Alle Angaben die zum Lesen der DRI erforderlich sind sind in der DRD enthalten.

Da in dieser Arbeit bisher in keinster Weise die Durchsetzung der Rechte beschrieben wurde, kommt nun das Kapitel über Sicherheit.

## 8.4 Sicherheit

Dieses Kapitel soll sich mit der Durchsetzung der auferlegten Beschränkungen für den Nutzer befassen und gleichzeitig mögliche Angriffe aufzeigen.

### 8.4.1 Vorbemerkungen:

#### Hard- oder Software

Auch Stadelmann beschäftigt sich in seiner Abhandlung [4] mit der Frage ob eine Hard- oder Softwarelösung effizienter ist. Er kommt zu dem Schluß, dass die Entscheidung für eine Hard- oder Softwarelösung keine Auswirkungen hat, da sich die Methoden des Reverse Engineering auf beide Arten gleich anwenden lassen. In diesem Punkt wird in dieser Arbeit ein anderer Standpunkt vertreten. Hardware-Reverse-Engineering setzt ausgefeilte Maschinen voraus, die in der Lage sind, Mikrocontroller bis ins Detail zu untersuchen. Diese entziehen sich im Allgemeinen dem Zugriff der breiten Masse. Reines Emulieren der Eingaben zu bestimmten Ausgaben reicht nicht aus, da es sich bei der Hardware um Schaltnetze handelt, bei denen die Ausgaben zu bestimmten Eingaben unmöglich vorhersagbar sind. Ein System auf Hardwarebasis kann also deutlich sicherer sein.

#### Breaking one Instance

Um mit einem DRM-System System nachhaltige Sicherheit zu gewähren muss sichergestellt werden, dass bei Kompromittierung einer Instanz das Gesamtsystem weiterhin funktionieren kann. Das Kompromittieren einer Instanz bedeutet in diesem Fall, dass entweder ein Inhalt ohne das entsprechende Recht betrachtet werden kann bzw. dass der digitale Inhalt von seinem DRM-Schutz befreit werden konnte. Ansätze zur Lösung dieses Problems werden im Verlauf des Kapitels vorgestellt. Auch für das Erstellen einer ungeschützten Kopie sind Stragien entwickelt worden, um zumindest den Ursprung dieser Kopie ausfindig machen zu können. Vergleiche dazu Watermarking und Fingerprinting weiter unten in diesem Kapitel.

#### Online vs. Offline

Grundlegend sind diese zwei Methoden zur Rechteverwaltung denkbar. Online in diesem Zusammenhang soll bedeuten, dass zur Authentifizierung eines Users eine Verbindung zu einem zentralen Server vonnöten ist. Offline hingegen würde bedeuten, dass alle Informationen, die man zum Betrachten des Inhalts braucht, lokal auf dem System des Nutzers

vorhanden sind. Während die Offlinevariante mehr Komfort für den Nutzer bietet, z.B. dass Nutzen der Inhalte auf mobilen Geräten ohne Zugang zum Internet (Autoradio), ist das Onlineverfahren für manche Anwendungen unabdingbar, zum Beispiel wenn eine Lizenz beinhaltet, dass ein Inhalt nur begrenzt häufig abgespielt werden kann. Auf Offline-Systeme wären hier zum Beispiel Replay-Attacken denkbar.

#### Authentifikation

Für alle mit DRM vorstellbaren Anwendungen muss sich der Benutzer gegenüber dem System authentifizieren. Der beste Schutz ist wirkungslos, wenn ein Angreifer einfach seine Identität nach Belieben ändern kann und so in die Rolle des legitimen Nutzers schlüpft.

### 8.4.2 Schutz der Daten

Notwendig, aber nicht hinreichend ist eine kryptische Komponente, die das Lesen der Rohdaten erschwert beziehungsweise ganz verhindert. Vorzuziehen sind hier mit Sicherheit bekannte symmetrische Verschlüsselungsverfahren. Da der Schlüsselaustausch dazu über öffentliche Netze ablaufen muss, sollte dieser über ein entsprechendes asymmetrisches Verfahren geregelt werden.
Obwohl nach dem Kerckhoff'schen Prinzip die Qualität einer Verschlüsselung ausschließlich vom Schlüssel abhängen darf und nicht von der Geheimhaltung des Verschlüsselungsverfahrens, nutzen viele Systeme eingene proprietäre Methoden. Dies nicht ohne guten Grund. Wenn ein potentieller Angreifer im Besitz sowohl des Verschlüsselungsverfahrens als auch des Schlüssels gelangt, liegem ihm die Daten offen. Das Bekanntsein von Schlüssel und Verfahren ist aber unabdingbare Voraussetzung für das Betrachten des Inhalts. Da dem User also prinzipiell alle benötigten Komponenten zur Verfügung stehen, soll ihm das Auslesen so schwer wie möglich gemacht werden.

### 8.4.3 Schutz des Systems

#### Code Encryption

Um das Reverse Engineering schwieriger zu machen, wird zum Beispiel Code Encryption verwendet. Dabei wird der gesamte Programmcode nur verschlüsselt gespeichert. Die Entschlüsselung erfolgt zur Laufzeit und immer nur in kleinen Teilen. Angriffe darauf haben schon stattgefunden. Literatur hierzu findet man im Bereich der Hacker- bzw. der Virusszene.

#### Code Obfuscation

Programme zur Code Obfuscation funktionieren prinzipiell ähnlich wie Code-Optimierer, nur dass hierbei nicht auf Geschwindigkeit und Effizienz, sondern auf Unverständlichkeit des erzeugten Codes hingearbeitet wird. Die Funktion des Programms wird dabei

nicht beeinträchtigt, jedoch das Nachvollziehen deutlich erschwert. Dabei wird auf drei verschiedenen Ebenen gearbeitet:

- Lexikalisch - Namen von Variablen und Bezeichnern werden unkenntlich gemacht, sodass die Bezeichnungen keine Rückschlüsse auf die Funktion zulässt. Dies erschwert die Arbeit des ReverseEngeneers und verzögert die Analyse.
- Datenorientiert - Man versucht Datentypen durch andere Datentypen zu ersetzen. So könnte ein Zahlenwert als Differenz angegeben werden oder eine boole'sche Variable durch die Gleichheit zweier anderer Variablen.
- Strukturell - Das Programm wird um Kontrollstrukturen erweitert, die mit konstanten Variablen gesteuert werden. Im späteren Programmverlauf macht dies keinen Unterschied, allerdings erzeugt dies beim Reverse Engineer Verwirrung.

**Polymorphie**

Polymorphie beschreibt eine Technik, bei der reduntante Codefragmente benutzt werden. Zum Beispiel können Hash-Werte auf zwei unterschiedliche Weisen berechnet werden, wobei jedoch sicherzustellen ist, dass sich die gleichen Werte ergeben. Ähnlich der strukturellen Code Obfuscation dient dies zur potentiellen Verwirrung und Überlastung des Reverse Engineers.

**Temper Checking**

Hiermit soll sichergestellt werden, dass sich das Programm in seinem Ursprungszustand befindet. Dabei gibt es verschiedene Möglichkeiten.

- Der Vergleich des Programmcodes mit einem vorher gespeicherten Hashwertes. Bei der Veränderung des Programms unterscheidet sich der aktuelle vom gespeicherten Wert.
- Der Vergleich bekannter Zwischenergebnisse von Subprozessen. Eine Abweichung von einem vorher bekannten Zwischenergebnis deutet auf eine Manipulation des Codes hin.
- Es wird überprüft, ob ein Debugger oder ähnliches im Hintergrund läuft. Dies wird bei modernen Betriebssystemen immer schwieriger, da beispielsweise nicht mehr auf Interrupttabellen zugegriffen werden kann. Die Implementierung dieser Methode muss plattformabhängig erfolgen, was das Verwenden deutlich erschwert.

Sollte eine dieser Methoden Alarm auslösen, wird der Programmfluss sofort oder zu einem späteren Zeitpunkt unterbrochen, oder aber das ganze System wird zum Stillstand gebracht. Diese Methoden dienen wie die vorher beschriebenen zum Erschweren des Reverse-Engineering-Vorgangs.

**Software Unification**

Wie bereits angesprochen ist es wichtig, dass bei Komprommitierung einer Instanz nicht das gesamte System betroffen ist. Das heißt, die Methoden, die zum Brechen der Sicherheitsvorkehrungen in einem System verwendet worden sind sollen bei einem zweiten System nicht zum Ziel führen. Um dieses Ziel zu erreichen wird Software Unification angewand und jedes Exemplar der Software ein wenig verändert.
Die Software Unification kann auf zwei Arten erfolgen: Zum einen intern, zum anderen extern. Extern bedeutet, dass sich auch die Funktionalität des Programms ändert, bei der internen Umsetzung ändert sich nur die Art und Weise der Ausführung. Während die externe Methode potentiell mehr Sicherheit bietet, verkompliziert sie auch das Management der Inhalte. Ein für ein bestimmtes System bereitgestellter Content ist damit nicht mehr unbedingt auf einer anderen Plattform abspielbar.

Ein weiterer Angriffspunkt auf ein solches System besteht darin, die zu irgend einem Zeitpunkt digital und unverschlüsselt auf dem System des Nutzers vorliegenden Inhalte abzugreifen. Davor kann nur ein sehr restriktives Betriebssystem und eine entsprechende, unkompromittierte Hardware schützen. Hierzu vergleiche Microsoft Media DRM im Abschnitt 'Projekte'. Zur Sicherstellung der Integrität des Systems ist es jedoch notwendig, dem User schon im Umgang mit der Hardware gewisse Restriktionen aufzuerlegen. Freien Betriebssystemen würde es auf diese Art und Weise unmöglich gemacht werden, DRM-geschützte Inhalte abzuspielen (dass es jemals DRM-Endsysteme auf Grundlage freier Software geben wird, bezweifelt der Autor stark).

All diese Methoden werden den entschlossenen, fachlich versierten Raubkopierer nicht aufhalten können. Sie dienen der Erschwernis, sind aber keinesfalls unknackbar. Wie bereits erwähnt, liegen Schlüssel und Verfahren den Benutzern nach erfolgreicher Analyse der Software offen. Wir können also nicht uneingeschränkt verhindern, dass Inhalte illegal kopiert werden. Um aber dennoch ein gewisses Ma§ an Schutzwirkung erreichen zu können gibt es Methoden zur Nachvollziehbarkeit des Ursprungs.

### 8.4.4 Nach dem Diebstahl

**Watermarking & Fingerprinting**

Watermarking und Fingerprinting nutzen prinzipiell ähnliche Verfahren, um Informationen in den Medien zu verstecken. Diese Informationen beziehen sich beim Watermarking auf den Urheber, beim Fingerprinting auf den User.
Für das Fingerprinting bedeutet das, Content müsste für jeden User einzeln mit einem Fingerprint versehen werden, beziehungsweise es muss beim Herunterladen der Software ein vorher eingebauter Fingerprint auf den User registriert werden. Auch wenn prinzipiell die Nachvollziehbarkeit gewährleistet ist, kann es in dem Moment zu Problemen kommen, wenn der Inhalt ohne das entsprechende Recht weitergegeben wird. Das heißt: User A lädt sich den Content herunter und gibt ihn an User B weiter, der seinerseits versprochen hat, sich das Recht zur Nutzung anderweitig zu verschaffen (zum Beispiel könnte dies geschehen, um User B Bandbreite zu sparen). Wenn User B das Versprechen nicht hält und

den Content wiederrechtlich nutzt, wird die Spur fälschlicherweise (!) zu User A führen. Gemäßdem Rechtsgrundsatz 'In dubio pro reo' kann User A - zumindest in Deutschland - dafür nicht rechtlich belangt werden. Derzeitige Watermarking Verfahren sind noch sehr anfällig für Beschädigung oder leichte Veränderung des Inhaltes. Das Fraunhofer Institut arbeitet momentan an einem Verfahren, dass sich bereits im Rundfunk bewährt hat. Bei einem Versuch war es möglich die meisten Wasserzeichen nach der analogen Ausstrahlung und anschliessender Redigitalisierung zu rekonstruieren.

## 8.5 Projekte

Diese Arbeit beschäftigt sich vorwiegend mit Mechanismen und Projekten hinter DRM. Deshalb sollen an dieser Stelle einige Projekte rund ums DRM vom eigentlichen DRM-System bis hin zur Software, die DRM-Technik nutzt, beschrieben werden.

**InterTrust Technologies**

ist eine Firma, die ein DRM-System entwickelt. Nach eigenen Angaben halten sie 32 US-Patente die Soft- und Hardwaretechniken abdecken und welche benutzt werden können um DRM und Trusted Computing zu implementieren.

**Microsoft Windows Media DRM**

ist nach Microsoft-Angaben eine bewährte Plattform um Inhalte zu schützen und sicher zu übergeben. Da Microsoft als führendes Mitglied der Trusted Computing Group (TCG) an der Entwicklung des Trusted Computing Modules (TCM) stark beteiligt ist, selber ein auf TCM aufsetzendes Betriebssystem entwickelt und ein eigenes DRM-System vermarktet, bieten sie wohl das umfassendste erhältliche DRM-System an.

**Der Apple iTunes Music Store**

ist ein Online-Musikvertriebssystem, welches DRM-Features aufweist. Trotz der scheinbar großen Akzeptanz des iTunes Music Stores von Seiten der Content-Industrie besitzt iTunes eine gravierende Schwäche. DRM-geschützte, mit der iTunes Software gebrannte Musikstücke können wiederum mit iTunes ins MP3-Format konvertiert werden, wobei sie ihren Schutz verlieren.

**Phillips und Sony,**

die beide scheinbar mit ihren eigenen DRM-Systemen scheiterten, sind nun bei InterTrust Technologies eingestiegen. Das, obwohl Sony's Absatzzahlen seitdem kopierschutzlosen Verkauf von Musik-CDs nach einer längeren Durststrecke wieder gestiegen sind.

**Open Rights Definition Language (ORDL)**

ist eine in der Open Software Community entwickelte Rights Definition Language, die wie viele andere auch auf XML basiert. ORDL ist von der Open Mobile Association (OMA) zum Standard für mobilen Content erhoben worden.

**Extensible Rights Markup Language (XRML),**

ebenso XML-basiert, ist die Wahl des Open eBook Forum (OeF) und wird auch bei Microsoft für ihr DRM verwendet. Dem Open eBook Forum gehören zum Beispiel Adobe, Microsoft und etliche eReader-Produzenten an.

## 8.6 Befürchtungen

Auch wenn dieses Thema in bisherigen wissenschaftlichen Publikationen eher stiefmütterlich behandelt wurde, möchte ich einige Risiken der DRM-Entwicklung aufzeigen. Durch eine Monopolisierung auf dem Markt der DRM-Systeme besteht in unserer heutigen Informationsgesellschaft das Risiko einer deutlichen Verzerrung des Marktes. DRM ist nicht nur für Inhalte wie Bilder, Musik und Videos gedacht. Auch eine Anwendung zum Beispiel auf Nachrichten wäre vorstellbar. In diesem Modell wäre es leicht realisierbar gewissen Gruppen den Zugang zu Nachrichten zu verwehren. Die Gruppenzuordnung unterliegt in diesem Szenario keinerlei Beschränkungen und obliegt nur dem Monopolinhaber. Finanzielle, politische oder ethnische Kategorisierung ist hier nicht auszuschließen. Die nicht notwendige Erfassung von personenbezogenen Daten ist in Deutschland zwar durch das Grundgesetz und einen Leitsatz des Bundesverfassungsgerichtes vom 15. Dezember 1983 geschützt, jedoch ist dies de facto kein Hindernisgrund detaillierte Interessenprofile der Anwender zu erstellen. Der Anwender ist hier gezwungen, dem DRM-Anbieter weitgehend zu vertrauen, sofern er sich denn überhaupt dieser Gefahr bewusst ist. Abhilfe würde hier nur eine vertrauenswürdige, zum Beispiel staatliche Stelle liefern, der die Verwaltung der jeweiligen Rechte obliegt. Denkbar würde dafür in Deutschland zum Beispiel die GEMA. Hierin würde die GEMA auch eine sinnvolle Aufgabe für die Zukunft finden.

## 8.7 Ein abschliessendes Wort

Diese Arbeit konnte nicht auf die Kontroversen Punkte in den Disskussionen um das Digital Rights Management eingehen. Mit dem Wissen das der interessierte Leser mitgenommen hat ist er jedoch in der Lage sich seine eigene Meinung zum Thema zu bilden und an Diskussionen teilzunehmen. Eine Prognose zu der zukünftigen Entwicklung um DRM ist noch nicht möglich, denn sehr entscheidend ist sind nicht nur die Vorstellung der Content-Industrie sondern vor allem auch die Begehren der Endanwender. Wenn sie durch DRM geschützte Produkte boykottieren ist die Industrie gezwungen ihre Pläne zu überdenken.
In jedem Fall ist eine massenwirksame, objektive Aufklärung über DRM nötig um auf demokratischem Wege eine Entscheidung für die Zukunft zu treffen.

# Literaturverzeichnis


[1] Iannella, Renato: DRM Architektures
*http://www.dlib.org/dlib/june01/iannella/06iannella.html*

[2] InterTrust Technologies: About Digital Rights Management
*http://www.intertrust.com/main/overview/drm.html*

[3] Unbekannt: Informationelle Selbstbestimmung - Was bedeutet das?
*www.datenschutz.de/recht/grundlagen/*

[4] Stadelmann, Thilo: DRM - Technik die Begeistert
*http://homepages.fh-giessen.de/ hg10013/Lehre/MMS/SS03/Stadelmann/text.htm*

[5] Stamp, Mark: DRM, The Technology behind the Hype
*http://home.earthlink.net/ mstamp1/papers/DRMpaper.pdf*

[6] LaMacchia, Brian A.: Key Challanges for DRM: An Industry Perspective
*Feigenbaum, Joan: Digital Rights Management, Washington, DC 2002*

[7] Commission of the European Parliament and of the Council - on measures and procedures to ensure the enforcement of intellectual property rights

[8] Gesetz zur Regelung der Urheberrechte in der Informationsgesellschaft

[9] Sietmann, Richard: Das Urheberrecht kennt kein Recht auf Privatkopie
*http://www.heise.de/ct/04/16/158/*

# Kapitel 9

# Standards für Biometrische Verfahren

*Stefan Mittendorf*

*Besonders nach dem Anschlag vom 11.09.01 ist Biometrie in den USA ein aktuelles Thema. In unserer Zeit wächst mehr und mehr der Wunsch nach Sicherheit im öffentlichen oder privatwirtschaftlichen Bereich. Die Biometrie soll hierzu verstärkt eingesetzt werden um vor ungewollten Zugriff und Zutritt zu schützen. Biometrie wird vorrangig in Bereichen, die sicherheitskritisch sind oder in denen sich viele Personen, die gegeben falls Identifiziert werden müssen aufhalten, eingesetzt. Diese Arbeit gibt erst eine Einführung in biometrische Systeme. Verfahren wie Fingerabdruck-, Handgeometrie-, Gesichts-, Iris- und Retinaerkennung werden genauer analysiert und verglichen. Um biometrische Systeme interoperabel zu gestalten werden Standards benötigt, diese werden im zweiten Teil der Arbeit vorgestellt und analysiert. Dabei stellt der Standard der BioAPI im zweiten Teil die zentrale Rolle da.*

## Inhaltsverzeichnis

## 9.1 Einführung in die Biometrie

Die Diskussion über die Einführung von Pässen mit biometrischen Merkmalen zeigt die Aktualität dieses Themas. Besonders nach dem Anschlag vom 11 September 2001 kam weltweit der Wunsch nach mehr Sicherheit auf. Biometrie bietet hier ein sicheres und leicht zu verwaltendes Verfahren um in sicherheitskritischen Bereichen Identitäten von Personen zu prüfen und festzustellen.
Zusätzlich bietet Biometrie eine gute Alternative zu dem zurzeit vorrangig verwendete Authentifizierungsverfahren die Anmeldung über Passwörter. Die hohe Anzahl an Passwörtern stellt für den einzelnen Benutzer ein Problem da. Da generell die Passwörter von einander unabhängig sein, nicht zu einfach sein, öfters geändert werden und nicht notiert werden sollten.

### 9.1.1 Begriffsabgrenzung

Der Begriff Biometrie (griech. Bios = Leben, Metron = Maß) bezeichnet die Lehre von Mess- und Zahlenverhältnissen der Lebewesen und ihrer Einzelteile sowie der Lebensvorgänge. Die Begriffe Biometrie und Biometrik werden oft synonym verwendet, doch gibt es Ansätze mit Biometrik die Wissenschaft und mit Biometrie das Messverfahren zu bezeichnen [6].
In dem Kontext der Computer-Sicherheit spricht man von einem Identitätsnachweis von Personen unter Verwendung ihrer individuellen, physikalischen (passiven) und verhaltensorientierten (aktiven) Merkmalen, die maschinell verarbeitet werden.
Allerdings müssen die Identifikationsmerkmale bestimmten Eigenschaften genügen um darauf biometrische Verfahren anwenden zu können.

- Universalität: das Merkmal muss bei jedem Menschen vorhanden sein
- Einzigartigkeit : bei jedem Menschen ist das Merkmal verschieden
- Beständigkeit: ohne Veränderung über die Zeit
- Erfassbarkeit: durch ein technisches System quantitativ messbar

Merkmale wie Fingerabdruck, Handgeometrie, Gesichtsform, Iris und Retina sind physische Merkmale, die diese Bedingungen erfüllen. Bei den Verhaltensmerkmalen werden Stimme, Handschrift und Gangart für biometrische Verfahren herangezogen.

### 9.1.2 Anwendungsbereiche und Einsatzgebiete

Die typischen Anwendungsbereiche sind zum einen **kommerzielle Anwendungen** die z.B. in Firmen genutzt werden. Ein klassischer Anwendungsbereich ist die **juristische Anwendung** von Biometrie z.B. Vaterschaftstest, DNA-Analysen, diese sind allerdings

sehr genau um möglichst keine Fehler zu machen. Der Einsatz von Biometrie von Seiten der **Regierung** ist relativ neu und wird auch noch diskutiert. In diesem Bereich verspricht man sich das Passwesen zu erneuern und die Überprüfung von Einwandern, Arbeitserlaubnissen oder Führerscheinen zu erleichtern [4]. Die Einsatzgebiete die man sofort mit Biometrie verbindet sind zum einen die *Benutzerzugangssicherung*, diese wird in sicherheitskritischen Bereichen wie beim Militär oder Gefängnissen genutzt, und zum anderen die *Gerätezugangskontrollen*, hierbei handelt es sich um die Authentifizierung zu Rechnern oder Netzwerken, die entweder vertraulich Daten enthalten z.B. Firmennetzwerk oder die kostenkritisch sind wie z.B. Handys, Geldautomaten.
Zu dem zeichnen sich immer neue Anwendungsgebiete ab in denen Biometrie einsetzbar ist. Ein gutes Beispiel hierfür ist der *Elektronische Zugang zu Dienstleistungen* z.B. E-Banking und E-Commerce, bei denen man sich vor einer Transaktion gegenüber den Kommunikationspartnern identifiziert.

### 9.1.3 Funktionsweise

Generell besteht ein biometrisches System aus einem *Sensor*[1] einer *verarbeitenden Einheit* [2] und einer *Datenbank*. Der Sensor soll die Messdaten einer Person aufnehmen und an die verarbeitende Einheit weiterleiten können. Diese wiederum kann die wichtigsten Informationen aus den Messwerten extrahieren und macht sie maschinell „verständlich“. Aus diese extrahierten, maschinell verständlichen Daten wird ein *Template*[3]erstellt. Hierbei handelt es sich um ein Datenpaket nach einer Dokumentvorlage. Anschließend wird dieses Template in der Datenbank gespeichert oder mit einem Referenztemplate aus der Datenbank verglichen. Die Datenbank selbst kann in den unterschiedlichen Formen realisiert sein. Ein Ansatz ist eine zentrale Datenbank, die leichter angreifbar ist und von Datenschützern abgelehnt wird, oder der Ansatz einer dezentralen Datenbank. Eine besondere Form ist das Speichern der Daten zu jeder Person auf einer Smart Card und diese dem Benutzer auszuhändigen.

Um eine Identität zu erkennen müssen vorerst Referenzdaten vorhanden sein. Dazu muss sich jeder Nutzer an dem System registrieren, dieser Vorgang heißt **Enrollment**[4]. Hierzu nimmt der Sensor die Daten der Personen auf (eventuell mehrmalige Messung) bis er ausreichend gute Messergebnisse bekommen hat. Aus diesen Messdaten wird ein Referenztemplate für diese Person erstellt und durch Verschlüsselung gesichert in der Datenbank gespeichert.

Biometrische Systeme werden in zwei Verfahrensweisen eingesetzt, der Verifikation und der Identifikation. Zur **Verifikation** überprüft das System ob eine angegebene Identität mit der wirklichen übereinstimmt. Dabei wird aus den Messdaten ein Template erstellt das mit dem Referenztemplate, dieser Identität aus der Datenbank, verglichen wird (1:1 Vergleich).

---

[1]Sensor:(engl. physical device), dieser kann je nach verwendeter Technologie anders ausfallen

[2]Verarbeitende Einheit: besteht aus einer Recheneinheit und Software

[3]Template:(engl. Formatvorlage, Muster im biom. Bereich) ein Record der wichtigsten Daten

[4]Enrollment:(eng. enrollment = einschreiben oder registrieren) bezeichnet die Lern- und Trainingsphase des Systems

Bei der **Identifikation** stellt das System die Identität einer Person fest. Das erstellte Template wird mit allen Templates der Datenbank verglichen und den am besten passenden aus der Datenbank (1:n Vergleich) zugeordnet und so dessen Identität ermittelt.

Das Verfahren des Vergleichens wird als **Match** bezeichnet, dieses kann auch je nach dem wie es eingesetzt wird in *Verify-Match* oder *Identify-Match* unterschieden werden. Die Ergebnisse eines Match sind prozentuale Übereinstimmung zwischen den aktuellen Messdaten und dem Referenztemplate.

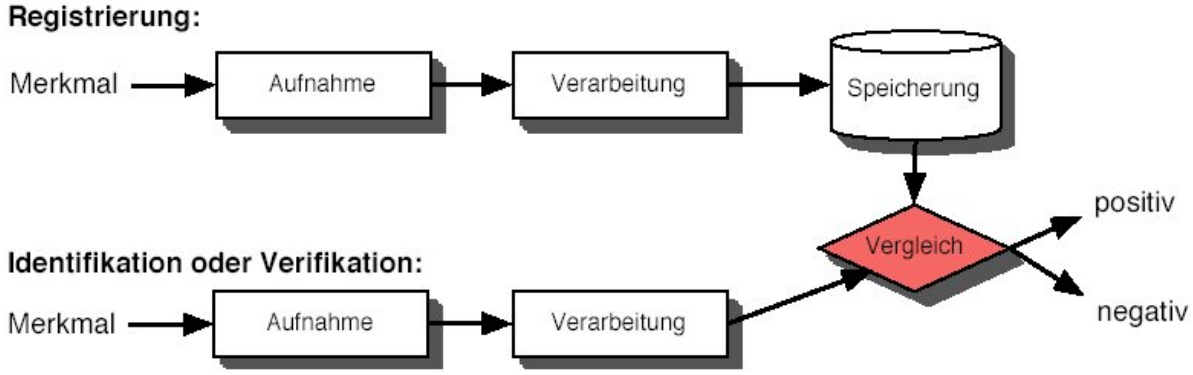


Abbildung 9.1: Schematische Dastellung des Enrollment, Vrifikation und Identifikation (aus [4],S.1)

### 9.1.4 Treffer- / Fehlerraten und Entscheidungsprozess

In biometrischen Systemen gibt es eine Vielzahl von Fehlerquellen. Die Ursachen sind äußert vielseitig, dennoch sollen hier die wichtigsten vorgestellt werden.
Biometrische Systeme speichern aus Kapazitätsgründen nicht die kompletten Messdaten, stattdessen nur bestimmte Informationen, die daraus resultierende Informationsreduktion kann zu Fehlern führen. Schon die Messdaten können fehlerbehaftet sein zum einen wenn der Sensor ungenau ist oder durch Umwelteinflüsse zum Messzeitpunkt z.B. Lichtverhältnisse, Hintergrundgeräusche bei Stimmanalyse oder Temperaturverhältnisse die den Sensor beeinflussen können. Menschliche Verhaltensmerkmale weisen eine bestimmte Varianz auf, z.B. kann ein Verhalten aufgrund von Emotionen beeinflusst sein. Des Weiteren können sich die Merkmale kurzfristig oder dauerhaft ändern z.B. durch Alterung oder Verletzung. Diese Ursachen machen eine 100prozentiege Übereinstimmung der aktuell gemessen Templatedaten und dem Referenztemplate fast unmöglich. Für den Vergleich wird daher ein Schwellwert hergenommen, um eine positive oder negative Entscheidung fällen zu können.

Um Fehler beschreiben bzw. einschätzen zu können hat man bestimmte Begriffe und Maßzahlen eingeführt. Ausgehend von den Mengen der registrierten und unregistrierten Benutzern will man wissen ob das System eine richtige oder falsche Entscheidung getroffen hat. Dabei sind die Trefferraten die Zurückweisung unbefugter Benutzer **CRR (richtige Zurückweisung)** [5] und die Akzeptanz registrierter Benutzer **CAR (richtige Akzep-**

[5]CRR:(engl. correct rejection rate = korrekte Zurückweisungs-Rate)

**tanz)**[6].

Im Umkehrschluss sind die Fehlerraten einmal die Akzeptanz unbefugter Personen **FAR (falsche Akzeptanz)** [7] , diese sollte in *sicherheitskritischen Systemen*[8] gleich Null sein. Der zweite Fehler der passieren kann ist, das ein registrierter Benutzer abgewiesen wird **FRR (falsche Zurückweisungs-Rate)** [9], eine hohe falsche Abweisungsrate widerspricht dem Grundsatz der Bequemlichkeit, da öfters frustrierende Neuversuche unternommen werden müssen.
Aus den Betrachtungen dieser Fehler ist ersichtlich das falsche Akzeptanz und falsche Abweisung vom Schwellwert abhängig sind. Sie sind auch umgekehrt proportional zu einander. Steigt die falsche Zurückweisungsrate sinkt gleichzeitig die Rate der falschen Akzeptanzen, solch eine Konfiguration würde mehr *Sicherheit* schaffen. Der Ansatz in solchen Systemen ist der, dass man lieber mehr Authentifizierungsversuche ablehnt aber dafür keinen unerlaubten Zugriff oder Zugang hat. Dem entsprechend ist der Schwellwert sehr hoch und die Authentifizierung kann schon durch kleine Fehler abgelehnt werden.
Umgekehrt kann man durch eine niedriger falsche Abweisungsrate und hohe falsche Akzeptanz Rate die *Bequemlichkeit* erhöhen. Hierbei gilt es über den Schwellwert einen dem Einsatz des Systems abhängigen Kompromiss zwischen Sicherheit und Bequemlichkeit zu finden.

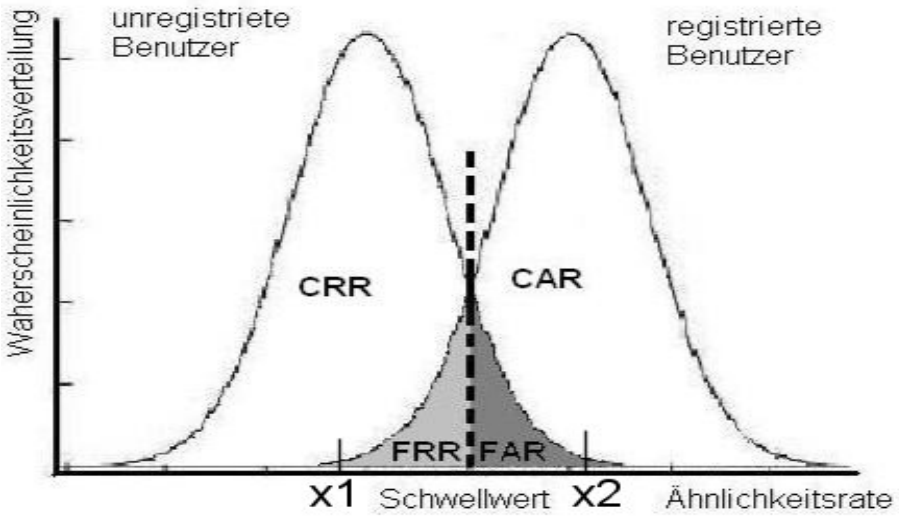


Abbildung 9.2: Diagramm zur Entscheidungsfindung

Das oben dargestellte Diagramm zeigt den Zusammenhang zwischen richtigen und falschen Akzeptanzen bzw. Zurückweisungen und den Schwellwert. Die Ähnlichkeit mit dem Referenztemplate ist auf der x-Achse dargestellt, d.h. rechts ist die Ähnlichkeit zwischen dem aktuellen Probetemplate mit dem Referenztemplate am größten. Der Schwellwert wird als Entscheidungspunkt benutzt, alles links davon wird abgelehnt und alles rechts davon

[6]CAR: (engl. correct accept rate = richtige Akzeptanzrate)
[7]FAR:(engl. false accept rate = Falsche Akzeptanz Rate)
[8]Systeme wie in denen die Existenz von Mensch und Maschine in Gefahr ist z.B. Gefängnis, Atomkraftwerk
[9]FRR:(engl.false reject rate = falsche Abweisungs-Rate)

akzeptiert.
Als Graph sind die Wahrscheinlichkeitsverteilungen der registrierten (FRR+CAR) und unregistrierten Nutzer (FAR+CRR) aufgetragen. Bei der Ähnlichkeit x1 ist die Wahrscheinlichkeit der korrekten Zurückweisung am höchsten und bei x2 ist die Wahrscheinlichkeit der richtigen Akzeptanz am höchsten. Der Verlauf des Graphen ist vom System abhängig. Neben dem Schwellwert beeinflusst das biometrische Verfahren und der Sensor die Trennschärfe des Gesamtsystems (z.B. Stimmanalyse ist wesentlich ungenauer als Iris-scaning). Daher würde der Iris-scan eine wesentlich niedrigere Fehlerrate aufweisen.

### 9.1.5 Angriffe auf biometrische Systeme

Bei biometrischen Systemen gibt es mehrere Angriffspunkte, da wäre zum einen der Angriff auf den Sensor oder Angriff auf die Kommunikation (zB. Replay-Angriffe) oder Templatedatenbank. Bei **Angriffe auf den Sensor** durch Diebstahl von Merkmalen registrierter Benutzer kann man das System angreifen. (z.B. nachgebildet Fingerabdrücke, Bilder für die Gesichtserkennung oder Einüben einer Handschrift).
Bei den meisten System reicht es aus zusätzlich einen Lifetest, also die Überprüfung ob es wirklich eine lebendige Person ist, zu machen um diese Angriffe abzuwehren.
Der **Angriff auf die Kommunikation** geschieht nach den Üblichen Verfahren der Netzwerkangriffe, die schon im Kapitel zu IPv6 als „Bedrohungen“ erklärt werden. Um ein System vor unsicherer Übertragung zu schützen kann man kryptographische Verfahren und MAC oder digitale Signatur verwenden (z.B. X9.84). Zusätzlich kann man Mechanismen einbauen in den sich Sender und Empfänger gegenseitig autorisieren.
Um **Angriffe auf die Templates in der Datenbank** zu erschweren sollte man die Datenbank vor unbefugten physischen Zugang schützen und die Templates verschlüsselt abspeichern.

Nachdem die Biometrie nun allgemein erläutert wurde, wird der nächste Abschnitt einige biometrischen Verfahren vorstellen und diese Vergleichen.

## 9.2 Biometrische Systeme

### 9.2.1 Fingerabdruck und Handgeometrie

**Fingerabdrücke**
Fingerabdrücke gelten als völlig einzigartig. Sie sind die ältesten Verfahren. Die erste Arbeiten mit Fingerabdrücken stammen aus dem 17. Jahrhundert. Um Fingerabdrücke zu Vergleichen gibt es 2 Verfahren: das *Pattern Matching* (Vergleich des gesamten Graubildes) oder das häufiger verwendete Verfahren das Vergleichen von *Minuzien/ Kleinigkeiten* (z.B. endende Täler, Verzweigungen, Schweißsporen, Fingeroberfläche).
Im Vergleich zu den kriminaltechnisch angewandten Methoden, in den man die Gesamtbilder also alle 10 Finger (250KByte pro Finger) speichert, werden in der Biometrie Koordinatenfelder besonderer Punkte als Template abgespeichert so reduziert man die Größe

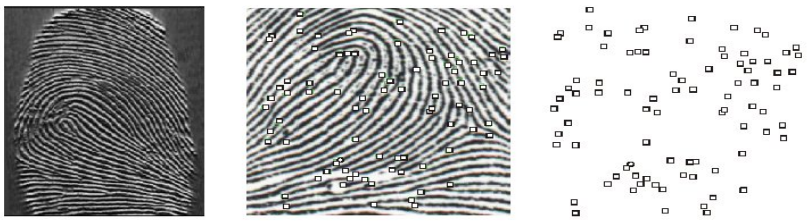


Abbildung 9.3: Generierung eines Tamplets aus einem Fingerabdruck
(aus [5], S. 11)

auf (250Byte - 1KByte) [2].
Schwierigkeiten stellen das Verdrehen des Fingers bei unterschiedlichen Messungen, die Verformung durch den Sensorkontakt da, allerdings sind diese leicht zu beheben. Um die Messdaten aufzunehmen wurden bis heute vier verschiedene Scannertyper entwickelt. Das älteste dieser Verfahren ist die *optische Aufnahme*, da diese Technologie auf optischen Aufnahmen basiert führen Aufnahmebedingungen und Feuchtigkeit der Haut zu Fehlern. Eine andere Technologie *CMOS Kapazitätsaufnahme* bedient sich der Ladungsunterschiede auf der Haut die mit Halbleitern also CMOS gemessen werden. Jedoch können Umgebungsladungen oder Hautfeuchtigkeit, die das Ladungsverhältnis verändert, zu Verfälschungen führen.
Ein viel versprechendes Verfahren ist die *Thermische Aufnahme*, bei der Temperatursensoren ein Wärmeprofil der Haut erstellen. Diese Technologie ist unabhängig von der Hautfeuchtigkeit.
Die der Zeit beste Technologie im Bereich des Fingerabdrucks stellt die *Ultraschall-Aufnahme* da. Der einzige Nachteil ist der Zeitaufwand.

Generell ist zu sagen das Fingerprintsystem sehr bequem und Platz sparend sind. Deshalb kann man sie dort nutzen wo andere Systeme zu groß wären z.B. bei Handys, Smart Cards und Computermäusen.

**Handgeometrie**
Ab dem Alter von 20 Jahren verändert sich eine Hand nur noch gering und bereits der

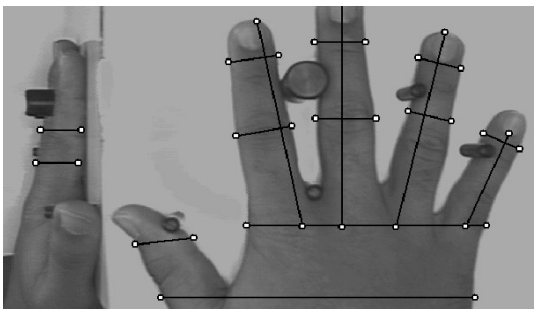

Abbildung 9.4: Messpunkte eines Handgeometrieverfahrens
(aus [2], S. 27)

Schatten einer Hand gilt als einzigartig. Hierbei werden zur Zeit 90 Parameter der Hand ermittelt, die Handoberfläche und Hautporen gehen hier nicht ein.

Der große Vorteil besteht in den kleinen Templates: 10-20 Bytes. Aber gegenüber den Fingerprint sind diese Verfahren unbequemer, da die Hand eine bestimmte Haltung einnehmen muss, zusätzlich braucht man zur Dickemessung aufwendige 3-dimensionale Optiken, was diese Systeme auch groß macht.

### 9.2.2 Gesichtserkennung

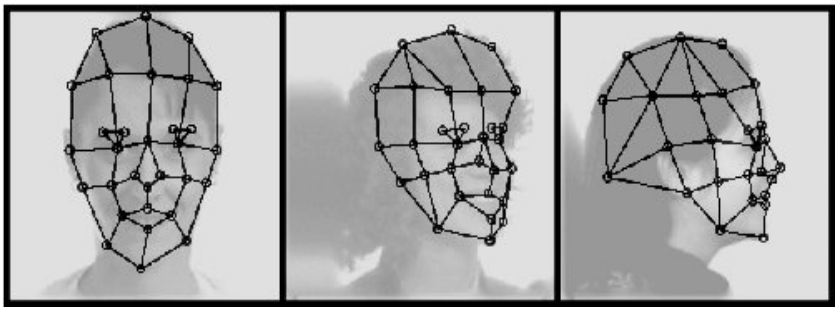

Abbildung 9.5: Gesichtserkennung nach geometrischer Merkmalsuntersuchung (aus [3])

Die Gesichtserkennung ist ein Verfahren das jeder Mensch manuell macht, indem er seine Mitmenschen automatisch an den Gesichtern erkennt. Maschinell gibt es mehrere Techniken eine Gesichtserkennung vorzunehmen, die drei wichtigsten sind:

*Template Matching:*
Bei dieser Technik wird das Gesicht in 100-125 kleineren Graustufenbildern abgespeichert. Diese Graustufenausschnitte werden über das aktuellen Template-bild gelegt und verglichen. Aufgrund der hohen Ungenauigkeit wird dieses Verfahren mit anderen biometrischen Verfahren angewandt.

*Untersuchung geometrischer Merkmale:*
Hier wird das Gesicht so aufgenommen das der Rechner daraus eine 3-dimensionale Form macht, um diese nach geometrischen Merkmalen zu vergleichen. Das größte Problem bei diesem Verfahren ist die Mimik, deswegen werden sich nicht durch Mimik verändernde Bereiche zur Messung benutzt (z.B. obere Kante der Augenhöhlen, Bereiche um die Wangenknochen, Augenabstand, Nase-Kinn-Abstand und die Seitenpartie des Mundes). Ein verwendetes Hilfsmittel sind bei diesem Verfahren Gitternetze. Dieses Verfahren ist zwar genauer als das Tamplte Matching, aber auch aufwendiger.

*Analyse per Fouriertransformation:*
Bei dieser Technik wird das Bild des Gesichtes nach Frequenzanteilen analysiert. Hierzu wird das Bild in eine eindeutig dazugehörende Frequenzdarstellung transformiert und dann verglichen. Der Vorteil ist das diese Frequenzbilder leichter zu vergleichen sind und die Genauigkeit durch die Anzahl der verwendeten Frequenzwerte beliebig verändert werden kann.

Neben den genannten gibt es noch einige andere Verfahren wie z.B. die Erfassung der Wärmeverteilung im Gesicht. Bei der Gesichtserkennung sind die Templates ca. 1.300 Bytes groß. Wie oben erwähnt trägt Mimik aber auch Alterung, Schminke, Brillen, Bartwuchs und Kopfbewegungen, sowie aufnahmebedingte Faktoren (Licht, Blickwinkel) zu Fehlern bei.

### 9.2.3 Iris und Retina-Scan

Die **Iris** ist absolut einzigartig sogar für die Augen einer Person, da die Irismuster nicht durch die DNA vorgegeben sind sondern sich während des 8. Schwangerschaftsmonats zufällig bilden. Die für die Biometrie wichtigen Parameter der Iris sind die Corona, Krypten, Fasern, Flecke, Narben, radikale Furchen und Streifen, wo hingegen die Farbe keine Rolle spielt. Das Irismuster ist auch stark resistent gegenüber Veränderungen. Alterung, Drogen- und Alkoholkonsum, selbst Augenkrankheiten wie Grüner Star und Augeninnendruck beeinflussen das Irismuster nicht. Lediglich Schädigung der Hornhaut ziehen erhebliche Veränderungen nach sich und es ist eine Neuregistrierung erforderlich. Zur Erfassung der Irismuster werden Schwarz-Weiß-CCD-Kameras verwendet. Durch die Schwarz-Weiß-Aufnahme und Reduzierung der Daten versucht man äußere Einflüsse wie Beleuchtung, Blickwinkel und Verformung der Iris durch unterschiedliche Pupillengrößen zu vermeiden. Diese Methode ist äußerst genau und so gut wie fehlerfrei. Die Wahrscheinlichkeit das 2 zufällig ausgewählte Bilder als übereinstimmend erkannt werden beträgt 1 : 7 Milliarden [6]. Die Templategröße liegt hier bei 512 -2048 Bytes.

Die **Retina (oder Netzhaut)** gilt als einzigartig und verändert sich nur durch Krankheiten. Gemessen werden derzeit 400 charakteristische Punkte. Äußerste Sicherheit gegenüber Attrappen, da ein implizierter Lifetest durch das Messverfahren vorgegeben ist. Hierdurch wird das Überlisten dieser Verfahren fast unmöglich. Nachteilig ist die aufwendige Spezialtechnik. Zu dem ist dieses Verfahren relativ unbequem da das Auge 1-2 cm Abstand zum Sensor haben muss und ruhig gehalten werden muss. Das System ist teuer und hat eine hohe Rückweisungsrate. Zusätzliche Probleme bereiten Kontaktlinsen.

Der große Nachteile von Verfahren die Iris und Retina vermessen ist die schlechte Benutzerakzeptanz, da die Nutzer Schädigung der Augen durch den Laser befürchten und da man ca. 2 Sekunden direkt in die Messeinrichtung hineinsehen muss.

### 9.2.4 Weitere Verfahren

Neben den oben erwähnten Verfahren gibt es noch eine Vielzahl anderer Verfahren. Hier wird jedoch kein Anspruch auf Vollständigkeit erhoben. Diese seien hier nur genannt: Bei der Handschriftanalyse kann man Personen über die Parameter Schreibdruck, Schriftbild und Geschwindigkeit zuordnen. Die Stimme kann über Tonhöhe, Dynamik und Wellenform analysiert werden. Neben der Handgeometrie kann man auch die Handoberfläche mit Poren und Adern für biometrische Verfahren nutzen. In Zukunft könnten Geruchsidentifikation, DNA-Analyse, Tastendruckdynamik, Gefäßmuster, Ohrgeometrie, Gangart, Blutbahnen für biometrische Verfahren genutzt werden.

### 9.2.5 Hybride Formen

Durch koppeln der biometrischen Systeme erhält man eine höhere Genauigkeit und man hat alternative Verfahren falls man bei bestimmten Personen ein Merkmal nicht messen kann. Um biometrische Systeme zu koppeln benötigt man vorgegebene Schnittstellen, von denen einige im nachstehenden Teil erklärt werden. Die Nachteile liegen in den großen Datenmengen, den Mehrkosten an Merkmalsmessung und Vergleichen. Oft bietet sich eine Verknüpfung von Verfahren an ein Beispiel für ein solches System ist das von den Firmen BioID und DCS kombiniertes System die Stimm-, Lippenbewegung- und Gesichtserkennung koppelt. Sinnvoll wäre auch die Verknüpfung von Handgeometrie und Fingerabdruck.

### 9.2.6 Vergleich der Methoden

Ein genauer Vergleich der unterschiedlichen biometrischen Verfahren ist schwierig, da es keine Festgeschriebenen Evaluationsmethoden gibt und die unterschiedlichen Systeme Entwicklungsunterschiede aufzeigen. Die Vergleichsparameter sind Genauigkeit, Zuverlässigkeit, Empfindlichkeit, Robustheit, Einfachheit und Kosten. Allerdings sind diese Verfahren, wie in der Tabelle zu sehen ist, auf unterschiedlichen Forschungsständen und dies erhöht die Schwierigkeit sie Vergleichen zu können noch einmal erheblich.

| biometrisches Verfahren | Einsatzreife | Identifikatonsmöglichkeit | Templategröße | S. Kosten (in US-doller) | S. Typ | S. Größe | Fehlerrate in Proz. | Täuschungsanfälligkeit |
|---|---|---|---|---|---|---|---|---|
| Gesicht | mittel | ja | 96B-2kB | 50 | kontaktlos | klein | 10 | hoch |
| Iris | hoch | ja | 256B | <3000 | kontaktlos | mittel | 1 | mittel |
| Retina | hoch | ja | 96B | 2500 | Kontakt Kopf | mittel | 5-15 | sehr gering |
| Fingerabdruck | hoch | ja | 200B | 5-100 | Kontakt los | groß | 5-20 | hoch |
| Handgeometrie | mittel | nein | 8B | 500 | Kontakt | groß | - | hoch |
| Sprache | hoch | nein | 2-64kB | 5 | kontaktlos | sehr klein | - | gering - hoch |
| Schrift | mittel | nein | 200B | 300 | Kontakt | mittel | - | gering - hoch |
| Tastaturanschlag | mittel | nein | 512B | 0 | Kontakt | klein | - | mittel |
| DNA | mittel | ja | 40B | 6500 | Zellprobe | Labor | sehr gering | sehr gering |
| Gang/ Bewegung | gering | ja | - | 50 | kontaktlos | klein | - | gering |
| Geruch | gering | nein | - | - | Kontakt | mittel | - | sehr gering |
| Blutbahn | gering - mittel | nein | 300B | 100 | Kontakt | klein - mittel | - | mittel |

### 9.2.7 Marktübersicht über Hersteller biometrischer Systeme

Biometrik ist eine Technologie die sich im Aufstieg befindet. Zurzeit stellen die Systeme mit Fingerabdruck und Handgeometrie den größten Marktanteil da, doch werden sie mit dem Weiterentwickeln der Technologie langsam durch Augen- und Gesichtserkennung abgelöst.

Es gibt eine Vielzahl an Firmen die im Bereich der Biomtrik arbeiten und forschen. Darunter gibt es einige große Firmen wie IBM (USA) die Fast die ganze Palette der biom. Verfahren abdecken oder Siemens (DE) die im Bereich der Fingerabdrücke forschen und Systeme produzieren.
Es gibt aber auch kleinere Firmen die sich auf Biometrik spezialisiert haben, z.B. die deutschen Firmen Biometric Solutions, die sich auf Fingerabdrücke spezialisiert haben, und BioID, die im Bereich der Gesichts und Spracherkennung tätig sind. Diese Beispiele sind nur exemplarisch eine größere Übersicht (Stand: 3. Quartal 2001) kann man in [6] auf den Seiten 16 - 18 nachschlagen. Die meisten Firmen stammen aus dem USA, Europa (Deutschland, Groß Britannien) oder Asien (China, Japan). Auffallend ist bei diesem Markt das die meisten Firmen selbst Forschungsarbeit betreiben. Nach Schätzungen steigt der Gesamtumsatz mit biometrischen Systemen von 2002 mit 600 Mio. bis 2007 mit 4 Mrd. US-doller [6]. Das zeigt den Trend des Aufstiegs in diesem Bereich. Die Konkursrate im biometrischen Bereich ist im Vergleich mit dem gesamten IT-Bereich unterdurchschnittlich.

Um solche unterschiedliche Verfahren hybrid zu nutzen muss man sich auf bestimmte Architekturen und Datenformate einigen. Um ein paar wesentlich Standards vorzustellen soll der nächste Abschnitt dienen.

## 9.3 Biometrische Standards

Die Standards sollen die Schnittstellen definieren und den Datenaustausch unterstützen, um Möglichkeiten zur Austauschbarkeit und/oder Zusammenarbeit von Systemen und Algorithmen zu schaffen. Die Verwendung eines standardisierten APIs (API = Application Programming Interface, Programmier-Schnittstelle) im Bereich biometrischer Systeme ermöglicht u.a. einen leichten Austausch biometrischer Technologien und eine leichte Integration mehrerer biometrischer Systeme über dasselbe Interface.
Zurzeit gibt es noch keinen einheitlichen Standard doch das BioAPI ist auf dem Vormarsch. Die internationale Versinon BioAPI 2.0 ist stark anerkannt und oft setzen nationale Standards darauf auf. Die konkrete Standards sind das BioAPI die ein interoperationale Architektur zur Verfügung stellt, das Kommunikationsprotokoll BIP, Abstrakte Datentypen und generische Datenformate wie CBEFF und der darauf aufsetzende Standard X9.84 der für eine sichere Übertragung und Speicherung dieser Daten dient. Weiterhin gibt es zu den unterschiedlichen biometrischen verfahren Standards um z.B. Schnittstellen zwischen Sensor und der verarbeitenden Einheit zu definieren.

### 9.3.1 BioAPI

BioAPI (Biometric Application Programming Interface) ist eine standardisierte Anwenderprogrammier-Schnittstelle zur Integration biometrischer Systeme in Anwendungen.
Sie ist eine Schnittstelle zwischen Software und Architektur um zu gewährleisten, dass eine biometrische Applikation mit mehreren biometrischen Einheiten arbeiten kann, ohne

genau zu wissen wie jede Einheit funktioniert. Die BioAPI soll den Entwicklern und Programmieren von biometrischen Verfahren helfen.
Weltweit haben sich 1998 85 Firmen und Organisationen zu dem BioAPI Consortium zusammengeschlossen. Ziel der Arbeit ist es eine Schnittstelle zwischen den Systemen zu definieren. Die Arbeit soll mehrere Dinge realisieren. Zum einen währe das eine *Modulverwaltung* und ein *standardisierter Datenbankmanager* um die Hierarchie zu entwerfen und zu verändern. Dem Entwickler will man *abstrakte biometrische Funktionen* zur Verfügung stellen die er nutzen und auf die unterschiedlichen Einheiten aufteilen kann (besonders zw. Client und Server). Zusätzlich soll das Gesamtsystem eine standardisierte Fehlerbehandlung bekommen.

**Das BioAPI Modell**
Um die Funktionsweise der BioAPI zu verstehen muss man sich erst einmal mit den Begriffen Vertraut machen. Wie oben erwähnt unterstütz die BioAPI **abstrakte Funktionen**. Die einzelnen Funktionen sind *Capture, Process, Match, Enrollment, Verfication und Identification.* Der **Client** ist in der BioAPI das System mit dem biometrischen Sensor der die Daten erfasst und an dem sich der Nutzer befindet. Er erledigt die Messwertaufnahme (Capture), kann eventuell auch schon diese Daten verarbeiten (Process). Der **Server** steht an einer physisch für den Nutzer nicht erreichbaren Ort, hier wird die Authentifikation durchgeführt. Dies geschieht nach dem üblichen Verfahren des Templatevergleichs (Match). Wie in der Abbildung zu sehen, beruht die BioAPI auf einer hierarchischen Struktur. Auf oberster Ebene steht die BioAPI-Applikation, darunter liegt das BioAPI-Framework und ganz unten stehen die BiometricServiceProvider (BSP). Um zwischen den Stufen kommunizieren zu können unterstützt das BioAPI **Streaming** also einen kontinuierlichen Datenfluss zwischen den Komponenten und das **Callbackprinzip** eine generische Funktion die von einer anderen Funktion aufgerufen wird und zu dieser als Parameter übergeben wird. An oberster Stelle steht die **BioAPI Application**, diese kann alle darunter liegenden Komponenten (Framework, BSP) über Aufrufe (sog. calls) aufrufen. Solch eine Applikation kann z.B. eine Kontrollapplikation sein, die das ganze System steuert, oder eine GUI-Applikation. Auf der nächsten Stufe liegt das **BioAPI Framework**, es vermittelt zwischen der Applikation und den BSP. So stellt es für die Applikation mit Hilfe der BSPs die abstrakten Funktionen zur Verfügung. Das Framework dient nicht nur zur Kommunikation zwischen den Komponenten sondern verwaltet alle unter ihm liegenden Komponenten (BSPs, Datenbank) und unterstütz den Austausch und Einbau von Sensoren. Der **Biometric Service Provider (BSP)** ist direkt mit einen oder mehren Sensoren verbunden und enthält meistens auch eine Bedienoberfläche für die Benutzer. Die Datenbank ist auch mit den BSPs verbunden. So kann es sein das die Datenbank mit einem BSP verbunden ist (zentrale Datenbank) oder das Datenbankteile mit unterschiedlichen BSPs verbunden sind (dezentrale Datenbank aber auch Smart-Card). Um das Streaming und die Callbacks zu unterstützen sind mehrere Interfaces vorgegeben, zwischen BioAPI Applikation und dem Framework sind das *BioAPI interface* für *Aufrufe* von der Applikation zum Framework und das *Application Callback Interface* für die *Antworten* (sog. callback) [10] vom Framework zur Applikation. Zwischen dem Framework und den BSPs gibt es das *BioSPI Interface* für Aufrufe vom Framework zu den BSPs und für Antworten das *Framework callback Interface.* Die **BIR´s (Biometric Information Record)** sind die Datenformate der BioAPI und folgen den Common Biometric Exchange

[10] In der Spezifikation wird er Begriff „callback“ verwandt

File Format (CBEFF), welches später erklärt wird. Die Signatur ist optional, sie wird aus dem Header und den Daten berechnet. Der BIR Datentyp bietet eine Signatur und/oder eine Verschlüsselung an. Wenn eine Anwendung ein BIR benötigt, kann dies über 3 Verfahren der Zugriff erfolgen. Einmal kann man einen „handle“(Henkel) der bei der Erstellung eines BIR´s generiert wird, einen Schlüsselwert oder den BIR direkt übergeben.

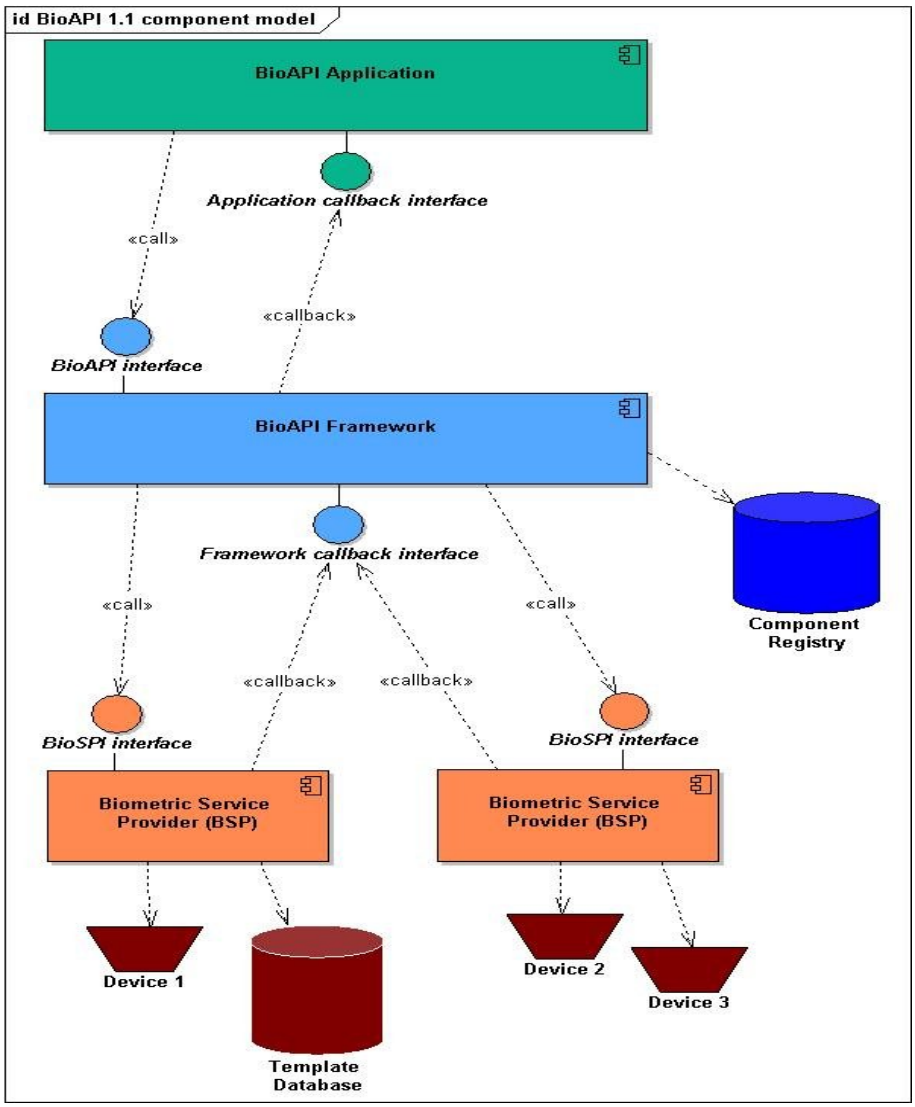


Abbildung 9.6: Hierarchie des BioAPI
(aus http://www.bioapi.org/)

### Funktionsweise des BioAPI

Die Verarbeitung läuft wie folgt ab.

Eine Verarbeitung (Enrollment, Verifikation und Identifikation) nutzt das Streaming-Interface um die BSP-Funktionen zwischen Client und Server auf zuteilen. Solch eine Verarbeitung kann von Client ausgehen, wenn sich ein Benutzer autorisieren will, oder vom Server ausgehend, wenn dieser eine Authentifikation erwartet, initialisiert werden.

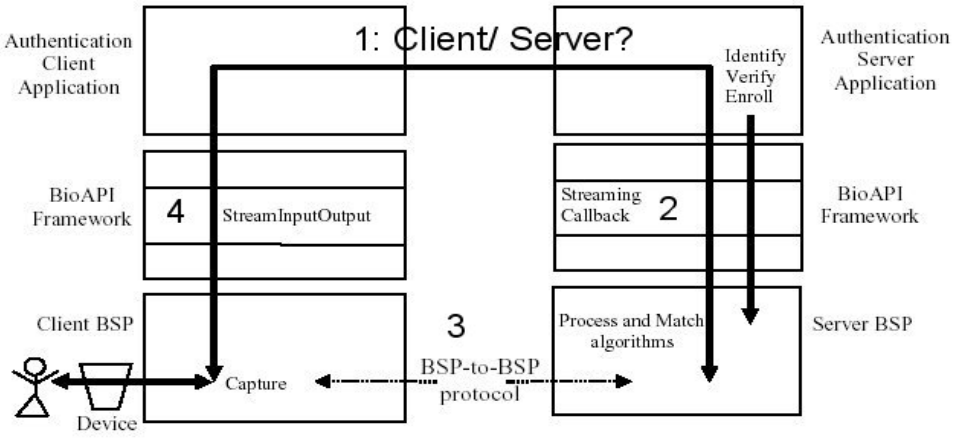


Abbildung 9.7: Grobablauf einer Verarbeitung
(aus [9], S.20)

Als erstes wird von der Client/Serverapplikation entschieden ob der Client oder Server die Verarbeitung übernimmt(1). Dabei wird der Kommunikationspartner der verarbeitende Komponene zur so genannten Partnerkomponente. Wenn z.B. der Server die Verarbeitungskomponente ist, ist der Client der Partner von dieser. Die Verarbeitungskomponente setzt ein Callback Interface zu seinem BSP (2). Daraus weiß der BSP das das System im Client/ Server Mode arbeitet und Initialisiert eine Kommunikation mit der Partner BSP. Die Anwendung ruft eine high-Level Funktion des Verarbeitungs-BSPs auf und dieser beginnt mit ein Streaming Callback zur Partner-BSP mit einem BSP-to-BSP Protokoll (3). Das Streaming Callback wird nur von dem verarbeitenden BSP benutzt. Wann immer dieser BSP mit einer Partner-BSP kommunizieren will, ruft er sein Streaming Callback Interface auf und kann so Nachrichten senden und Antworten erhalten. Die „StreamingInputOutput“ Funktion wird von der Partner-Applikation benutzt, um Nachrichten an die Partner BSP zu schicken und später dessen Rückgabe zu erhalten (4). Danach wird die Antwort der verarbeitenden Anwendung zugeschickt, so das diese mit den Daten arbeiten kann.

**Das Modulmanagement**

Bei der Installation, geben BioAPI Komponenten (Framework, BSP) und Sensorsysteme ihre Daten an das Modulmanagement. Die Applikation und das Framework können so schauen welche Komponenten mit welcher Leistungsfähigkeit installiert sind. Dies ist wichtig für die Applikation um eine dynamische Entscheidslogik bezüglich der zu verwendenden Komponenten zu ermöglichen. Außerdem erleichtert das Modulmanagement einen Ein-, Ab- und Umbau von Komponenten.

### 9.3.2 BioAPI 2.0

Wie aus der Versionsnummer ersichtlich ist dieser Standard eine Weiterentwicklung der BioAPI 1.1. Es ist der internationale Standard der BioAPI.

Die BSP der BioAPI 2.0 wurden in kleinere Einheiten aufgeteilt und unterschieden. Hier besteht ein BSP aus: BioAPI unit, Sensor unit, Archiv unit, Matching algorithem unit und Processing unit.
Zusätzlich wurde Architektur um eine Komponente erweitert den **BioAPI Biometric Function Providers (BFP)**, dieser ist hierarchisch unter den BSPs zu finden. Er bietet Funktionen zur De-/ Installation und Verwaltung von BioAPI-Komponenten wie Sensoren an.

### 9.3.3 BIP (Bieometric Interworking Protocol(ISO/IEC24708))

Das BIP legt hauptsächlich die BioAPI Framework-zu-Framework Kommunikation fest. Bei dieser Anwendung ist das Protokoll der BioAPI entsprechend implementiert um die BioAPI Applikationen und BSPs zu unterstützen. Das BIP kann auch dazu genutzt werden um eine BioAPI Applikation auf einem PC und die Framework mit ihren BSPs auf mehreren PCs laufen zu lassen. Hierbei können die Funktionen capture, verfification, identification, enrollment usw. von einer entfernten Applikation genutzt werden.

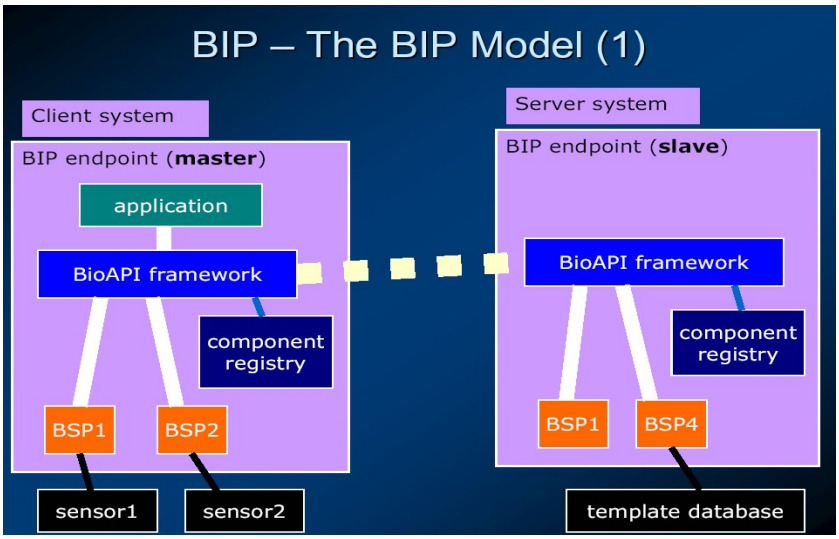


Abbildung 9.8: BIP-Architektur
(aus [7])

Ein Vorteil der sich daraus ergibt ist das eine biometrische Einheit die auf einem System installiert ist von einem anderen System genutzt werden kann. So erlaubt das BIP eine einfache, geschlossene Stand-alone-implementation von einer entfernten BioAPI Applikation genutzt zu werden. Zusätzlich erleichtert sich der Austausch von biometrischen Laufwerken und BSPs, da die zentrale Applikation nicht geändert werden muss. Weil die Applikation normal mit dem Framework kommuniziert und die anderen Frameworks für sie transparent sind.

### 9.3.4 Common Biometric Exchange File Format (CBEFF)

Standard zur Speicherung und Austausch von biometrischen Daten unterschiedlicher biometrischer Systeme. Dazu wird die Datei in 3 Teile: Header, Biometrie spezifischer Speicherblock und Signatur unterteilt.

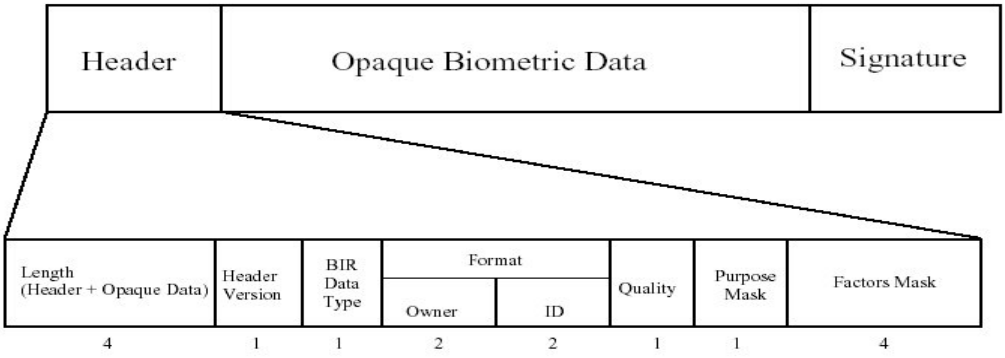


Abbildung 9.9: Eine BIR nach dem CBEFF
(aus [9], S. 16)

Im Header sind die Kontrollinformationen (wie Größe des Pakets, Owner von dem das Paket stammt und eine ID) für die jeweilige Datei abgelegt.
Im Biometrie spezifischen Speicherblock stehen die biometrischen Daten, sie setzen sich aus dem Template und eventuellen Zusatzinformationen zusammen. Zusatzinformation könnte z.B. ein Subheader sein, der Informationen wie Versionsnummer, Datenlänge, Verschlüsselungsinformationen und Ähnliches enthält. Wie die Daten dort konkret abgelegt sind ist offen und kann vom Benutzer, einer Arbeitsgruppe oder einen anderen Standard definiert sein.
Der Signaturblock ist optional und kann entweder Signaturinfos oder die MAC enthalten um die Integrität der Daten zur gewährleisten. Der CBEFF macht zwar eine Kommunikation zwischen den Komponenten möglich. Doch lässt der Klartext der biometrischen Daten Platz für Angreifer.

### 9.3.5 X9.84

Der X9.84 ist ein amerikanischer Standard er setzt auf dem CBEFF, der den Austausch von biometrischen Daten ermöglicht, auf und erweitert ihn im Bezug auf Sicherheit. Das Hauptproblem das mit dem X9.84 Standard gelöst werden soll sind Angriffe auf die Datenbank und die Kommunikation, die bereits in dem Abschnitt über „Angriffe auf biometrische Systeme“ vorgestellt wurden. Hierzu werden hauptsächlich Verschlüsselung, Kryptographie, Mac und Signaturen benutzt um die Vertraulichkeit der Daten zu gewährleisten.

In der Abbildung ist links das Datenformat nach CBEFF abgebildet, dieses wird schrittweise zu einen X9.84 entsprechenden Datenformat erweitert. Aus dem optionalen Signaturblock des CBEFF wird der „Integrity Block“. Dieser enthält Informationen über

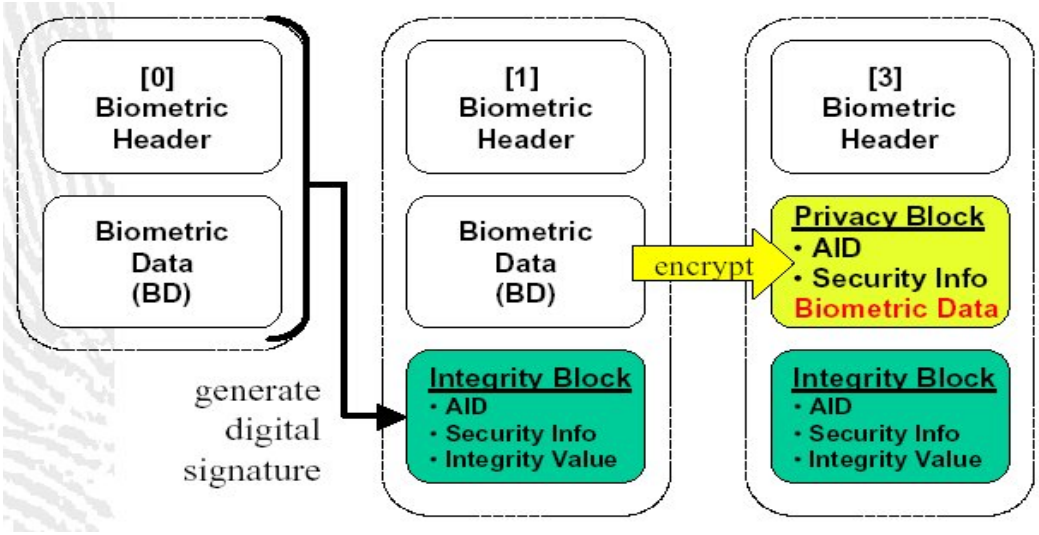


Abbildung 9.10: Blockbildung nach dem X9.84
(aus [11])

Verwendete Sicherheitsalgorithmen „AID“ [11](Verschlüsselung und MAC), „Security Info“ (Algorithmusparameter, Informationen zum Schlüssel) und „Integrity Value“ (digitale Signatur oder MAC).
Zusätzlich wurde aus den biometrischen Datenblock der „Privacy Block“, der neben dem „AID“ und „Security Info“ die verschlüsselten Daten enthält.
Des Weiteren wird ein X.509-Zertiffikat, das von einem „Trustcenter“ stammt, in den Datenblock eingekapselt.

### 9.3.6 BAPI

Das BAPI (Biometric API) ist ein dem BioAPI ähnlicher Standard, der von I/O-Software und Microsoft entwickelt wurde. Auch dieser Standard soll eine Interoperabilität von unterschiedlicher biometrischer Hardware sicherstellen.
Aus der Entwicklersicht stellt dieser Standard 3 Entwicklungsschichten zur Verfügung. Die höchste Ebene ist die dritte und die abstrakteste Ebene. Diese Schicht soll dem Entwickler ermöglichen schnell Prototypapplikationen zu entwerfen ohne ein umfangreiches Wissen über die technischen Details zu haben. Auf Ebene 2 kann der Entwickler Features des biometrischen Verfahrens anwenden (z.B. wenn er mit einem System bestehend aus Fingerprintern arbeitet, die Features die das Fingerprintverfahren liefern zu nutzen). Auf der technischen ersten Ebene soll der Entwickler die Features des konkret verwendeten Sensors nutzen können. Die BAPI stellt für die Entwicklung ein entsprechendes Framework zur Verfügung, das dem „Windows Device Developers Kit“ ähnlich ist.

Das große Konzept das hinter der BAPI steht ist das man jedem Sensor (als Quelle) Hardwarekanäle zur Verfügung stellt.[8] Mit diesen Kanälen kann man Daten generieren (Enrollment) oder einen Effekt bewirken (Abweisung/Akzeptanz). Wie in der Abbildung

[11] Algorithem Identifier

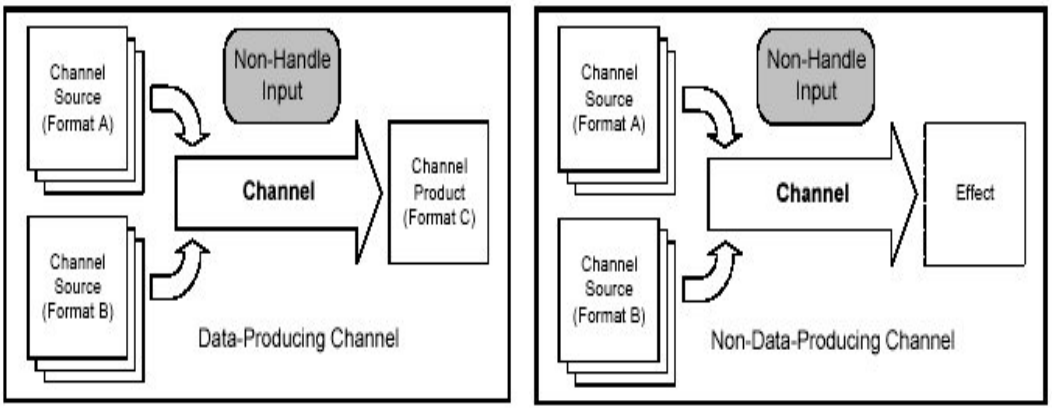


Abbildung 9.11: Kanalmodi der BAPI
(aus [8])

zu sehen können diese Kanäle mehrere Quellen haben. Dies könnte man gut für Matchverfahren anwenden indem man die aktuellen Messdaten als eine Quelle und das Refernztemplate als zweite Quelle nutzt. Der Vorteil den die BAPI bietet liegt darin das man die Features auf allen Ebenen nutzen kann [8], besonders auf der ersten Ebene können die Features von Sensoren durch die Standardisierung wegabstrahiert werden.

### 9.3.7 XML-Standards zum Datenaustausch

**XML Common Biometric Format (XCBF)** Das XCBF soll den Austausch und das Abspeichern der biometrischen Daten erleichtern um die Interoperanilität zu steigern. Der Standard bietet vollautomatische Methoden zur Personenerkennung. Der Vorteil gegenüber den binären Datenformaten ist das man XML fähige Systeme und Programme unterstützen kann. Der XCBF baut im Wesentlichen auf dem CBEFF auf und definiert einige sichere XML-Verschlüdsselung/Kodierung für den CBEFF. Um die Sicherheit (Integrität, autorisierten Ursprung) zu gewährleisten nutzt man die anerkannten XML Verfahren „Canonical XML Encoding Rules (CXER)“ oder den oben beschriebenen X9.84 Standard. Dieser Standard soll besonders in Web-spezifischen Anwendungen genutzt werden oder um die gegenseitige Übersetzung von dem in der BioAPI1.1 spezifizierten BIR in X9.84 zu realisieren. Um die Webanwendungen sicher zu machen existiert das „Web Services Security XCBF Token Profile“, diese unterstützen X.509 Zertifikate und CRL´s. Da der XCBF aus dem CBEFF hervorgegangen ist, hat man ein festes Format und man kann zwischen XML oder binären Format wechseln (z.B. bei Leistungsstarken Systemen XML und bei Leistungsschwachen Systemen oder Datenbanken das binäre Format).

### 9.3.8 Internationaler Überblick

Auf nationaler und internationaler Ebene sind mehrere Gremien damit befassen, Kriterien zu definieren um biometrische Verfahren evaluieren und Pilotprojekte vergleichen zu können. In Deutsachland stellt sich der Forschungs- und Standardisierungsstand wie folgt

dar. Allein die Forschung und Entwicklung biometrischer Systeme ausreichend zu analysieren ist äußerst schwer. Da zum einen aussagekräftige Informationen von den Firmen, die die meiste Forschungsarbeit betreiben, schwer zu erhalten sind und da es kaum öffentlich Forschungsarbeit in diesem Bereich gibt. In Deutschland gibt es eine große Zahl an Projekten doch die Zahl der involvierten Hochschulen ist sehr klein und überwiegend an Detaillösungen orientiert. Doch mit der Beteiligung deutscher Institute und Unternehmen an europaweiten Arbeiten steigt die Aktivität hierzulande. Deutsche Projekte sind: Bio Trust von BMWi, H204M von BMBF und BioIS vom Bundesamt für Sicherheit in der Informationstechnik. Als ein Beispiel für die Forschungsarbeiten und Forschungsstand soll ein konkretes Beispiel dienen. Eine Forschungsarbeit zur Evaluation von biometrischen Systemen, die an mehreren Fraunhofer-Instituten stattfand. Hier hat man 10 auf dem deutschen Markt erhältliche Produkte getestet. Hierzu wurden sie in diesem Institut installiert und von 40 Testpersonen ein halbes Jahr genutzt. Lediglich 2 Geräte wurden als alltagstauglich eingestuft.
Die USA ist zurzeit marktführend in der Biometric. Hier fördert das Biometric Consortium, bestehend aus 6 Ministerien die Entwicklung von Standards und Evaluationsverfahren.
Neben Europa und USA entwickelt man im Ostasiatischenraum an Biometrik. In Nordamerikanischen und asiatischen Raum gibt es eine Vielzahl an Aktivitäten im privatwirtschaftlichen und öffentlichen Sektor.
Nach diesem Überblick sollen im letzten Abschnitt biometrische mit anderen Authentifikationsverfahren verglichen werden.

## 9.4 Bewertung biometrischer Systeme

### 9.4.1 Vor- und Nachteile biometrischer Verfahren gegenüber herkömmlichen Methoden

**Vorteile**:
Ein Vorteil ist die Personenbezogenheit, man kann nicht nur unterscheiden ob eine Person für einen Bereich Zugang und Zugriff hat, sondern kann auch die Identität feststellen. Zudem gibt es dem Benutzer eine Bequemlichkeit, da er sich nichts merken braucht und bei der Identifikation nicht mal eine Eingabe in das System tätigen muss.
Der Diebstahl der Identifikationsmerkmale stellt sich im Gegensatz zu besitzorientierten Verfahren als äußerst schwierig oder technisch sehr aufwendig da.
Verfahren und Techniken zum Angriff auf den Bereich der wissensbasierten Authentifikation wie „Brute Force“ sind im biometrischen Bereich unmöglich.
Als Vorteil genannt sei auch noch der einheitliche Sicherheitslevel innerhalb einer Gruppe.

**Nachteile**:
Biometrie hat auch einige Nachteile, das heikelste Thema hierbei stellt die Benutzerakzeptanz da. Verfahren die viel Mitwirkung von den beteiligten Personen verlangen (z.B. Retina) werden als aufdringlich empfunden. Auf der anderen Seite werden biometrische Verfahren die unauffällig ablaufen (z.B. eine Kamera in einem öffentlichen Gebäude, die die Gangart von Personen aufzeichnet) als Eingriff in die Privatsphäre gesehen [6]. Zudem

gibt es Nutzer die biometrische Systeme aufgrund von Angst vor Gesundheitsschäden, kulturellen oder religiösen Gründen meiden [6].
Die Geschwindigkeit solcher Systeme stellt einen weiteren Nachteil da. Da hier nicht einfache Pins oder Passwörter sondern größere Datensätze verglichen werden müssen. Bei einer Identifikation steigt die Zeit der Auswertung mit der Anzahl der Nutzer und ist daher besonders Zeitintensiv. Zusätzlich verursachen große Nutzerzahlen eine höhere Fehlerrate, da die Templates bei größeren Nutzerzahlen nicht mehr so leicht voneinander zu unterscheiden sind.
Auch die Zuverlässigkeit und Sicherheit von biometrischen Systemen variiert sehr stark, so das einige Systeme sehr gut und andere nicht anwendbar sind.

### 9.4.2 Ausblick

Die G8 - Staaten Deutschland, Frankreich, Großbritannien, Italien, Japan, Kanada, Russland und USA wollen gemeinsam einen Pass mit biometrischen Elementen entwickeln und einführen. Nähere Ausführungen hierzu bietet der nächste Teil der Seminararbeit.
Ein Trend der jetzt bereits einsetzt ist das biometrische Verfahren PINs und Passwörter ablösen. Darüber hinaus bieten sich eine biometrische Kennzeichnung bei Kreditkarten und EC-Karten gerade zu an, da man genauer die Person überprüfen kann.
Schließsysteme wie z.B. Haustüren oder Autotüren könnten allein über Biometrie realisiert werden. Hierbei wäre auch eine „Biometrische Eintrittskarte“ denkbar. Das heißt, das herkömmliche Eintrittskarten (z.B. Flugtickets, Kino-, Theater- und Bahnfahrkarten) von solch einem System abgelöst werden. [6]
Ein gutes Anwendungsgebiet wäre die Heimautomation [12]. Hier könnten biometrische Sensoren die Bewohner erkennen und zum Beispiel das Licht so wie von der Person gewünscht einstellen oder wenn eine bestimmte Person das Haus betritt die Kaffeemaschine, Fernseher oder Computer anschalten.

[12]Systeme die aufgrund von Sensoren Aufgaben im Haus übernehmen können, z.B. die Heizung aufgrund von Temperatursensoren steuern.

# Literaturverzeichnis


[1] Demmler Markus, *Overview of Biometrics*,
Stand: 25.06.05, Universität Augsburg,
http://www.informatik.uni-augsburg.de/ kimjongh/biometrics/folien/overview.pdf

[2] *Biometrische Identifikation Sachstandsbericht*,
Büro füt Technikfolgen-Abschätzungen beim Deutschen Bundestag (TAB), Stand: 25.06.05,
http://www.tab.fzk.de/de/projekt/zusammenfassung/ab76.pdf

[3] Prof. Dr. Peter Gerhard, *Biometrische Erkennungsverfahren*,
Stand: 25.06.05,
www.hochschulverwaltung.de/tagung/ kassel/abstracts/folien/biometrie_kasselv2.pp

[4] Heckl Florian , *Der Körper als Passwort*,
Stand: 25.06.05, Universität Ulm,
http://www.informatik.uni-ulm.de/ni/Lehre/WS01/HS-Biometrische-Systeme/ausarbeitungen/SystemeStandardsEvaluation.pdf

[5] Dr. Bäumler Helmut, Gundermann Lukas, Dr. Probst Thomas, *Stand der nationalen und internationalen Diskusion zum Thema Datenschutz bei biometrischen Systemen*
Stand: 25.06.05, Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein, Kiel,
http://www.datenschutzzentrum.de/download/tabga.pdf

[6] Prof. Dr. Dornberger, Probst Fabian, *Biometrics ein Überblick*,
Stand: 25.06.05, Fachhochschule Solothurn,
http://www.fhso.ch/pdf/publikationen/dp03-w02.pdf

[7] Trigilia, Alessandro, *Bieometric Interworking Protocol(ISO/IEC24708)*,
Stand: 25.06.05,
http://www.csrc.nist.gov/pki/BioandEAuth/Presentations/ Wednesday, 20March 2030/Trigilia_new.pdf

[8] *Requirements and Anylysis The Biometric API Standard*,
I/O Software, Stand: 25.06.05,
http://www.iosoftware.com/Documents/The 20Biometric 20Standard.pdf

[9] *BioAPI Specification Version 1.1*,
BioAPI Consortium, Stand: 25.06.05,
http://www.bioapi.org/

[10] Tilton Catherine J. *Developing BioAPI and CBEFF-Compliant*,
Stand: 25.06.05,
http://www.biometrics.org/html/bc2002_sept_program/4_bc0061_TiltonBrief.pdf

[11] Stapleton Jeff *American National Standard X9.84-2001 Biometric Information*,
Stand: 25.06.05,
http://www.ncits.org/tc_home/m1htm/docs/m1020002.pdf§

[12] Hladik Robert, *Einführung in die Biometrie*,
Stand: 25.06.05,
http://www.informatik.uni-ulm.de/ni/Lehre/WS02/HS-Biometrische-Systeme/ ausarbeitungen/HLADIK.PDF

# Kapitel 10

# Verwendung von biometrischen Verfahren im Passwesen

*Oliver Münstermann*

*Das Pass- und Ausweiswesen hat eine sehr lange Tradition in der Geschichte der Menschheit. Dabei wurden bisher schon biometrische Merkmale, wie z.B. Aussehen(Bild), Unterschrift, Größe in den Ausweisdokumenten festgehalten und bei den Kontrollen verglichen. Die zunehmende Globalisierung und der stetig ansteigende internationale Personenverkehr, in Kombination mit der allgegenwärtigen Terrorbedrohung führen dazu, dass dieses manuelle Verfahren nicht mehr die gewünschte Sicherheit und Leistungsfähigkeit bietet. Demzufolge haben viele Regierungen beschlossen, zukünftig maschinenlesbare Ausweisdokumente mit prägnanteren Daten (biometrischen Kenngrößen) auszustatten und somit die Kontrollsicherheit zu erhöhen. Dieser Seminarvortrag stellt diese Thematik in einigen Teilbereichen dar und versucht den derzeitigen Stand der Entwicklung nahe zu bringen.*
*Dabei werden im 1. Kapitel die Grundlagen und Rahmenbedingungen kurz vorgestellt, warum das automatisierte biometriegestützte Passwesen von Regierungen und Behörden schnellstmöglich eingeführt wird. Nachfolgend werden die Einsatzgebiete, die nicht nur Grenzkontrollen betreffen, sondern auch Geheimdienste, Sicherheitsverwahrung und Firmen erörtert. Im folgenden Abschnitt werden die derzeit bevorzugten biometrischen Verfahren und Merkmale zur Identifikation in Einzelnen, für das Passwesen wichtigen Details, kurz erläutert. Das 4. Kapitel betrachtet die Ansätze zur Speicherung der durch biometrische Scans erhaltenen Daten zum Einen auf maschinenlesbaren Dokumenten mit integrierten Chips, als auch zum Anderen auf zentralen oder dezentralen Servern. Nachfolgend wird im 5. Kapitel eine Laufzeit- und Praktikabilitätsbetrachtung dieser Verfahren, beim Einsatz als Einzelscans als auch bei Scans von Menschenmengen, dargelegt. Den derzeitigen Stand der Einführung in den USA und Europa spiegelt das 6. Kapitel wieder. Im letzten Kapitel werden Risiken und Gefahren der biometriegestützten automatisierten Scanverfahren erläutert. Zudem wird auf rechtliche Bedingungen des Datenschutzes eingegangen. Die Schlussbetrachtung und der Ausblick auf kommende Entwicklungen schließen diesen Seminarvortrag ab.*

## Inhaltsverzeichnis

## 10.1 Einleitung und Motivation

Aufgrund der zunehmenden Globalisierung und des stetig wachsenden Passagieraufkommens im nationalen und internationalen Reiseverkehr, wird schon lange über moderne, einfache aber dennoch sichere Verfahren zur Identifikation von Personen nachgedacht. Hierbei versucht man Schritt für Schritt sich die moderne Scantechnik zu nutze zu machen. Die elektronische Identifikation von Personen und die Verifikation von Personalien, mittels biometrischer Eigenschaften, steht schon lange im Fokus der Forscher und Entwickler. Auf diesem Gebiet wurde schon im letzten Jahrtausend intensiv geforscht und Szenarien entwickelt, wie man die Technologie zeitsparend und gewinnbringend einsetzen könnte. Diese Entwicklung wurde durch den Terroranschlag vom 11. September 2001 auf das World Trade Center maßgeblich beschleunigt. Analysen über die Hintergründe und Beteiligten des Anschlages ergaben, dass die Anschläge von Personen durchgeführt wurden, die schon unter dem Verdacht und der Beobachtung der Regierungen standen. Eventuell hätte dieses Ereignis durch genauere Kontrollen und schärfere Restriktionen im Passagierverkehr verhindert werden können.
Die USA, unter der Regierung von Präsident Bush, verabschiedete als Reaktion auf die verheerenden Anschläge und der internationalen Bedrohungen noch im Jahr 2001 den so genannten **Patriotact**. Dieses Gesetzeswerk sieht neben vielen Regularien zur Kontrolle der Bürger Amerikas, auch die Kontrolle und Beschränkung von Ein- und Ausreisen anderer Staatsbürger vor. Hierzu wurde das Visa-Recht maßgeblich verschärft und an die neuen Rahmenbedingungen angepasst. Somit sind Einreisen von längerer Dauer (>90 Tage) nur noch mit offiziell beantragtem Visum möglich. Es wurde zudem festgeschrieben, dass bis **Mitte 2005** alle Reisenden mit maschinenlesbaren Reisedokumenten ausgestattet sein müssen und zudem bei der Einreise weitere biometrische Merkmale wie z.B. der Fingerabdruck eingescannt und mit Datenbanken abgeglichen wird.
Die Europäische Union, als auch viele andere Länder, wie Dubai und die Vereinigten Arabische Emirate beschlossen auch, schnellstmöglich ein stringenteres und besseres Kontrollwesen für den internationalen Passagierverkehr zu etablieren. Dabei wurden die von der US-Regierung vorgegebenen Umsetzungspläne als Meilensteine für die eigenen Umsetzungsvorhaben verwendet. Somit plant Deutschland die Einführung der ersten, verbesserten Reisepässe ab November 2005 und führt dazu schon Machbarkeitsstudien durch. Die Schweiz, als Nicht-Mitglied der EU, versucht auch diesem Trend zu folgen und plant die Einführung fürs 4. Quartal 2005. In Skandinavien und den Beneluxländern laufen schon Projekte zur Einführung der neuen Technologie.
Einige der größten Befürworter dieser Regelungen sind:

- die Amerikanische Regierung,
- die ICAO (International Civil Aviation Organisation),
- die IATA (International Air Transport Association),
- die Grenzkontrollorgane (FBI, BGS...),
- die jeweiligen Innenminister für Sicherheit.

## 10.2 Einsatzgebiete des automatisierten biometriegestützten Pass- und Ausweiswesens

Favorisierte Einsatzgebiete für das automatisierte Passwesen stellen Flughäfen, Bahnhöfe und Häfen dar. Das automatisierte Pass- und Ausweiswesen wird hierbei zunächst nur für den internationalen Reiseverkehr interessant, da zunehmend Staatengemeinschaften, wie die USA und die EU[1] ihre eigenen Grenzkontrollgesetze haben, z.B. das Schengener Abkommen. Hierbei haben Bewohner dieser Staatengemeinschaften innerhalb der Staatengemeinschaft ein fast unbeschränktes Reiserecht, ohne zusätzliche Kontrollen. In Verbindung mit dem internationalen Grenzverkehr werden von vielen Ländern Visa für die Einreise verlangt. Dieser Bereich könnte durch die Verwendung von automatisierten Verfahren wesentlich vereinfacht werden, da die Visa nicht extra geprüft werden müssten und nur in Verbindung mit dem Reisepass gültig sind. Dadurch könnten Verwaltungsaufwand, Zeit und Kosten gespart werden, unter gleichzeitiger Erhöhung des Komforts der Reisenden. Der wichtigste Aspekt der automatisierten biometrischen Kontrollen ist der erwartete Gewinn an Sicherheit durch die Verwendung dieser modernen und nahezu unbestechlichen Verfahren. Dadurch, dass eine Vielzahl der Reisedokumente zukünftig in elektronischer Form gespeichert und ausgestellt werden, kann hier der Missbrauch und die Fälschung begrenzt und somit die Kriminalität bekämpft werden.
Individuelle Lösungen für die biometrische Passkontrolle könnten allerdings auch bei Großveranstaltungen wie der Fußball-WM oder Olympiade, Anwendung finden. Überall wo größere Menschenmengen an gefährdeten Bereichen zusammenkommen und kontrolliert werden müssen, könnten diese Verfahren Einzug halten. Bei der Fußball-WM könnte die Kontrolle der Eintrittskarte, die an den Personalausweis gekoppelt ist, durch Abgleich mit den gespeicherten Daten auf dem Pass vereinfacht und sicherer werden. Der Schwarzmarkthandel sowie der Besuch unerwünschter Gäste würde somit beschränkt werden.
Ein weiterer Einsatzraum wäre aber auch die Verwendung der biometrischen Identifikation in sicherheitskritischen Bereichen, beispielsweise im Pentagon oder im Reichstag. Hier könnten die Kontrollen von Besuchern und Angestellten zu einer erhöhten Sicherheit führen und Anschläge oder Spionage wesentlich erschweren. Vorstellbar wäre zudem, dass die derzeitigen manuellen Identifikationsprozeduren, wie das PostIDENT[2]-Verfahren, nicht mehr von Menschen durchgeführt werden sondern, mittels biometrischer Identifikation, automatisiert werden.
Derzeitig verfügen schon einfachste Geräte, hierzu zählen vor allen Dingen PDA's, Mobil-Telefone oder aber auch Notebooks, über sehr gut funktionierende Scansysteme, um einen Zugang zum System nur ausgewählten und autorisierten Menschen zu gewähren.
Das Spektrum für einen Einsatz biometriegestützter Verfahren ist immens. Daher beschränkt sich diese Seminararbeit auf den Bereich des automatisierten Pass- und Ausweiswesens in Kombination mit biometrischen Verfahren.

[1] Europäische Union
[2] Post-Identifikation

## 10.3 Biometrische Verfahren im automatisierten biometriegestützten Passwesen

Derzeit existiert eine Vielzahl biometrischer Verfahren auf dem Markt, deren Einsatzspektrum sehr vielfältig ist. Für den Einsatz im automatisierten Pass- und Ausweiswesen können nicht alle Verfahren uneingeschränkt genutzt werden, da sie zum Teil zu ungenau oder zu aufwändig sind. Daher wurde von den internationalen Behörden ICAO[3] und IATA[4] eine Auswahl aus den gängigen Verfahren getroffen, die eine akzeptable Scan- und Identifikationsleistung aufweisen. Diese Verfahren werden hier nur kurz dargelegt:

### 10.3.1 Gesichtsscan

Der Gesichtsscan ist ein Scanverfahren, dass sich sowohl bei Einzelpersonen, als auch bei Menschenmengen einsetzen lässt. Voraussetzung hierfür ist der uneingeschränkte Blick des Sensors/Optik auf das zu betrachtende Objekt. Beim Gesichtsscan werden die Merkmale mehrerer Messpunkte erfasst und dann als Gesamtobjekt in einem Template gespeichert. Derzeit finden 2 Scanverfahren Anwendung:

- ⊙ Vollbild (Farbe / Graustufen)
- ⊙ Token-based (Abstände und Geometrien)

Bei einem **Vollbild-Template** ist nach Studien ein minimaler Speicherbedarf von 11 kB notwendig. Bei dem auf **Token basierenden Template** muss mindestens 9 kB auf dem Speichermedium verfügbar sein um eine ausreichende Identifikations- und Verifikationssicherheit zu gewährleisten [3, S.14ff]. Der Zeitbedarf (inkl. Auswertung) für einen Einzelscan liegt bei ca. 5 sek, wenn das Gesicht klar erkennbar ist und die entsprechenden Templatedaten vorliegen. Als Ziel für die Anwendung dieses Verfahren steht jedoch nicht der Einzelscan, sondern die Anwendung auf Menschenmengen. Hierbei sollen die Personen schon während ihrer Bewegung über das Flughafengelände, z.B. auf dem Weg von der Maschine zum Gepäckband gescannt und verifiziert werden. Dies spart Zeit und ist für die Personen mit keinerlei Aufwand verbunden.
Da dieses Verfahren derzeit noch anfällig gegenüber Veränderungen am Gesicht ist, hierzu zählen beispielsweise Bartwuchs und Schönheitsoperationen, bedarf es noch präziserer Einzelscanverfahren, um die nötige Scansicherheit zu gewährleisten.

### 10.3.2 Irisscan

Der Irisscan stellt derzeit die sicherste Nachweismethode für biometrische Daten dar, da die Wandlungen am menschlichen Auge in Bezug auf die Lebenszeit minimal sind. Brillen

[3] International Civil Aviation Organisation
[4] International Air Transport Association

und Kontaktlinsen stören das Scanergebnis nur, wenn sie den Blick auf das Auge modifizieren. Allerdings eignet sich dieses Verfahren nicht für den Einsatz bei Menschenmengen, da die Sensoren relativ nah am Auge sein müssen.
Der Irisscan erzeugt ein Template, dessen Größe nach ICAO-Standards zwischen 2100 Bytes und 52250 Bytes liegen kann, abhängig von der verwendeten Scan-Genauigkeit [8, S.23ff]. Für die Verwendung dieser Templates wurde vorgeschrieben, dass die Templates in einem austauschfähigen, verlustfreien Datenformat gespeichert werden müssen. Dies ist derzeit das RAW-Datenformat. Um die Datenraten allerdings gering zu halten, wird das Template mit der nahezu verlustfreien JPEG2000 Komprimierung reduziert [8]. Die Laufzeit eines solchen Scans hängt maßgeblich von der Geschwindigkeit der Justierung des zu messenden Objektes (Kopf und Auge) ab. Dies beträgt 5 - 10 Sekunden (10 - 30cm Abstand und gerader Blick in die Optik). Der Scanprozess dauert ca. 1 - 3 Sekunden. Somit kann von einer durchschnittlichen Mess- und Auswertezeit von ca. 10 - 15 Sekunden ausgegangen werden.

### 10.3.3 Finger / Handflächenscan

Der Handflächenscan scannt die wichtigsten Linien einer Handfläche. Einzelne Scanner kombinieren diesen Scan gleichzeitig mit dem Fingerabdruckscan und sparen somit Zeit und erhöhen die Sicherheit. Hierzu werden oftmals Kontaktscanner eingesetzt, die ein direktes Berühren der Scanner-Oberfläche erfordern. Dadurch ist das Verfahren bei Menschenmengen nur bedingt einsetzbar, da der Verschmutzungsgrad mit jeder gescannten Hand zunimmt. Schweiß, Unreinheiten und Kratzer auf der Optik sind hierbei die größten störenden Faktoren. Die Dauer für einen Handflächenscan liegt zwischen 1 - 10 Sekunden, abhängig vom verwendeten Scanverfahren (Handfläche inkl./ exkl. Finger). Die erstellten Templates liegen in der Größenordnung von mindestens 11kB (10 Finger je 1kB + Handgeometrie 1kB) [9]. Eine Komprimierung könnte hier bedingt zum Einsatz kommen, da die Daten im Vektorformat gespeichert werden.

### 10.3.4 Unterschriftenscan

Der Unterschriftenscan scannt die Unterschrift einer Person und vergleicht sie mit den Mustern, die die Person vorher abgelegt haben muss. Hierzu muss mit einem Spezialstift auf einem Tablet die Unterschrift geleistet werden. Diese wird dann mit dem Unterschriftentemplate anhand von bestimmten Merkmalen, wie z.B. Bogenschwung, Schriftlage, Länge der Buchstaben und Druck verglichen. Da Unterschriften mit ausreichend Übung fälschbar sind, stellt dieses Verfahren nur ein zusätzliches Verifikationsmedium dar, welches nur in Verbindung mit weiteren sichereren Methoden Anwendung finden darf.
Eine Möglichkeit der Kombination von zwei Methoden stellt die Verwendung von Unterschriften und gleichzeitigem Fingerabdruckscan dar [22]. Hierfür wurden neuartige Stifte entwickelt, die diese hybride Funktionalität bieten [vgl. Abb. 1.1] . Als Einsatzgebiet hierfür wäre beispielsweise der Check-In sehr geeignet, da somit anhand des Passes und der Unterschrift sowie dem Fingerabdruck, eine erste Verifikation der Personalien erfolgen könnte. Die Scandauer ist quasi proportional zur Unterschriftlänge. Das zu speichernde

Template könnte hierbei direkt vom Pass ausgelesen werden. Dies könnte durch einen Scan oder einer Datenübertragung via Funk geschehen.


Abbildung 10.1: PenOne
Quelle: [22]

### 10.3.5 Zusammenfassung und Vergleich

Man kann leicht erkennen, dass nur das Gesichtsscan-Verfahren für einen Einsatz bei Menschenmengen geeignet ist. Da die Scanleistung derzeit allerdings nicht ausreicht, kommt nur die Kombination von mehreren Scanverfahren in Frage. Vorzugsweise wären der Irisscan, in Kombination mit einem Handflächen/- Fingerscan anzusehen, da hier derzeit die höchste Scanleistung erzielt werden kann. Der Einsatz bei Menschenmengen muss daher noch verbessert werden. Da viele Systeme voneinander unabhängige biometrische Merkmale scannen, können sie oftmals problemlos parallel eingesetzt werden.
Die Verfahren basieren allesamt auf der Erstellung und Speicherung von Daten in Form von Templates. Diese sensiblen Daten müssen auf geeigneten Medien und in geeigneter Struktur gespeichert werden, um den Richtlinien der ICAO zu entsprechen.

## 10.4 Maschinenlesbare Ausweisdokumente und weitere Speichermedien für biometrische Daten

Die Verwendung von biometrischen Verfahren zur Identifikation und Verifikation bedingt neuartige Speichermedien für die gesammelten Daten. Hierbei muss zwischen lokalen Speichermedien die beim Endanwender (Reisepässe und Visa) und globalen Speichermedien (Datenbanken) unterschieden werden. Jedes dieser Speichermedien hat Vor- und Nachteile sowie wichtige technologische Rahmenbedingungen, welche Einsatzspektrum vorgeben.

### 10.4.1 Maschinenlesbares Reisedokument - Der neue Reisepass

Um biometrische Daten vergleichen zu können, müssen die zukünftigen Reisedokumente maschinenlesbar sein. Nur so lassen sich die zahlreichen biometrischen Daten auslesen und vergleichen. Hierzu wurde ein neuartiger Reisepass entwickelt, der die biometrischen Daten auf einem Chip speichert, welcher mittels Funktechnik ausgelesen werden kann. Dieses Verfahren wird schon heute, z.B. in der Wirtschaft zur Kennzeichnung von Produkten, angewendet. Das neue Reisedokument wird somit, zusätzlich zum dem in Plastik verschweißten Dokumentensatz über einen Elektronikanteil in Form eines RFID[5]-Chips und einer Leiterschleife verfügen [15] [vgl. Abb. 1.2] .

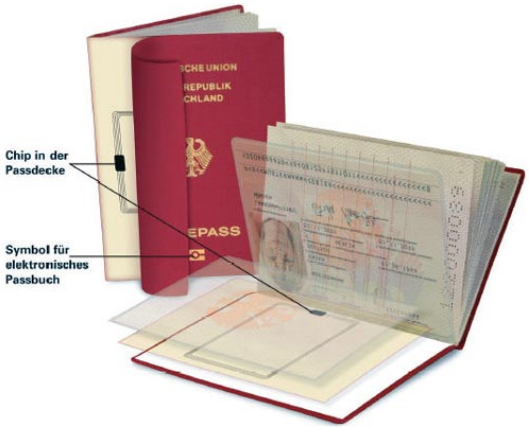


Abbildung 10.2: Elektronischer Pass
Quelle: Bundesdruckerei

Die Leiterschleife dient hierbei zum Empfang und zum Senden der Daten sowie zur Leistungsaufnahme für den Chip. Die RFID Technik stellt hierbei einen kostengünstigen Ansatz zur Datenspeicherung dar, da zum Einen die zivilen, freien Frequenzen (z.B. 125kHz oder 13,56MHz) zur Übertragung genutzt werden und zum Anderen die Daten auch in verschlüsselter und signierter Form auf einem relativ billigen Chip (ca. 1 Euro / Stück) abgelegt werden können, dessen Speicherkapazität derzeit schon für 2 biometrische Templates ausreicht. Zudem ermöglicht der Einsatz von kontaktlosen Ausleseverfahren eine hohe Lebensdauer der Dokumente und der Lesegeräte. Gleichzeitig wird das System für

[5]Radio Frequency Identification

den Kunden sehr einfach zu bedienen, wodurch Fehlverhalten und somit bedingter Zeitverlust vermieden werden kann. Der große Nachteil an diesem System ist jedoch die Funkübertragung, da dies eine universelle Auslesbarkeit des Chips mit sich bringt. Jeder, der über einen entsprechenden Transceiver[6] verfügt, kann theoretisch den Reisepass auslesen und die Daten speichern und auswerten. Allerdings sollen die Daten auf dem Chip verschlüsselt werden und zunächst mittels DES-Kryptoverfahren abgerufen werden können. Im weiteren Schritt soll der Chip in einer Private/Public-Key-Infrastruktur als auch mit Zertifikaten ausgestattet sein [15].
Als Alternative für die Speicherung der biometrischen Daten eines Fingerabruckes oder von Gesichtsmerkmalen kann auch ein Barcode-ähnlicher Aufdruck auf den Pass erfolgen [vgl. Abb. 1.3] . Dieser könnte dann mittels optischem Scanner und Fingerprintscanner ausgelesen und verglichen werden. Hierbei könnte auf die Verwendung der Funktechnik verzichtet werden. Dadurch wäre allerdings auch kein kontaktloses Verfahren zum Auslesen des Passes möglich. Dieses Verfahren stellt somit eher eine suboptimale Lösung dar, da sehr häufig reisende Menschen eine höhere Abnutzung des Reisepasses hätten. Da die Daten für die Verifikation nicht nur auf dem Ausweisdokument gespeichert sind sondern auch bei einer autorisierenden Stelle stellt sich die Frage, wie die Referenzdaten gespeichert werden.


Abbildung 10.3: Konventioneller Pass mit Finger-Scan-Barcode
Quelle: [23]

### 10.4.2 Biometrie-Datenbanken

Die Speicherung von biometrischen Daten auf zentralen Servern ist bekanntlich eine weitere Bedingung für die Nutzung des automatisierten biometrischen Passwesens. Dabei besteht derzeit bei den einführenden Ländern noch Diskussionsbedarf über den Speicherort und die Nutzungsbedingungen von biometrischen Daten. Dies ist weniger ein technologischer, als ein politischer und rechtlicher Streitpunkt.

#### Speicherung auf nationalen Servern

Hierbei werden die Daten auf lokalen Servern des entsprechenden Herkunftslandes gespeichert und können bei Bedarf von den autorisierten Instanzen abgerufen und verwendet,

[6] Transmitter-Receiver

jedoch nicht dauerhaft gespeichert werden. Dieses Verfahren hat den Vorteil, dass die personenbezogenen Daten nicht ohne Grund außer Landes kommen und so eine globale Überwachung durch einen einzelnen Staat beschränkt wird. Zudem kann der Missbrauch von Daten begrenzt werden, falls das System durch einen Angriff geschädigt werden sollte. Der Nachteil liegt zum Einen in möglichen Redundanzen und der aufwändigen Kontrolle ob eine Person mit zwei unterschiedlichen Identitäten im System ist und zum Anderen in der Kontrolle, ob die personenbezogenen Daten nicht doch von anderen Ländern gespeichert und weiterverwendet werden.

#### Speicherung auf internationalen Server

Die zentrale Speicherung bietet den Vorteil, dass die Daten für jedermann einheitlich gespeichert und verfügbar sind, so dass bei Kontrollen die Daten nur von einem zentralen Server abgerufen werden müssen und keine aufwändige Systemarchitektur erstellt werden muss. Redundanzen und Doppelanmeldungen können hier leichter vermieden werden. Der große Nachteil liegt in der Datensicherheit, denn das System ist durch die zentrale Verwaltung anfälliger für Missbrauch aller gespeicherter Daten, wenn die Sicherheitsmechanismen überwunden wurden. Zudem kann nicht gewährleistet werden, dass ein Staat alle Daten kontrolliert und dann für eigene Zwecke missbraucht.

### 10.4.3 Implantate

Eine weitere Möglichkeit bietet die Speicherung der Daten auf Chips, die unter die Haut gepflanzt werden. Diese Technik ist schon bei der Tieridentifikation im Einsatz, z.B. bei Hunden, Wildtieren etc. Hierbei werden die Daten auf einem gekapselten RFID-Chip gespeichert welcher unter der Haut sitzt und nur durch Geräte aktiviert und ausgelesen werden kann, die sich in ca. 0cm - 5cm Abstand befinden. Diese Methode stellt einen Eingriff in die persönliche Selbstbestimmung dar und ist somit zunächst abzulehnen, jedoch bietet es die Sicherheit, dass gewisse biometrische oder genetische Daten immer ausgelesen werden können, auch wenn der Reisepass nicht verfügbar ist. Diese Technik könnte im Bereich des Strafvollzuges und der Überwachung von Kriminellen zum Einsatz kommen.

### 10.4.4 Fazit der Speichermedien

Die Wahl der Speichermedien für die biometrischen Daten lässt derzeit einen großen Spielraum zu und bedarf noch der ausführlichen Diskussion der involvierten Länder um einen einheitlichen Standard zu schaffen. Dabei stehen oftmals eher nationale Interessen und die nationale Rechtsprechung im Vordergrund, als ein kooperatives Miteinander. Daher wird dies wahrscheinlich ein langwieriger Prozess werden, bevor ein einheitlicher Standard gefunden ist.
Jedoch sollen die Daten in Deutschland lokal gespeichert bleiben [26], um den derzeitigen Datenschutzregularien zu entsprechen. Jede dieser Speicherungsarten hat allerdings auch Auswirkungen auf das Laufzeitverhalten der Verifikationsprozeduren.

## 10.5 Laufzeitverhalten vs. Praktikabilität

Im täglichen Einsatz des automatisierten Passwesens spielen Leistungsfähigkeit und Durchsatz der Scan- und Auswerteverfahren eine maßgebliche Rolle. Dabei muss unterschieden werden, ob ein Verfahren für Einzelpersonen oder Gruppen geeignet ist. Nach internationalen Standards wurde festgelegt, dass mindestens zwei biometrische Verfahren [12] und Analysen angewendet werden müssen, um die geforderte Sicherheit zu gewährleisten. Da jedes dieser Verfahren Zeit in Anspruch nimmt, müssen die Kontrollanlagen dementsprechend dimensioniert werden oder die Zugriffsmethoden auf die Referenzdaten optimiert werden. Derzeit benötigt man für einen Datenzugriff auf ein Irisscan-Template in einer Menge von 1 Millionen Templates ungefähr 1 Sekunde [27]. Dies ergibt rechnerisch, bei einer Bevölkerungsanzahl in Deutschland von 80 Millionen, eine Zugriffszeit von 80 Sekunden. Bei 1 Milliarde Menschen wären dies schon 1000 Sekunden. Daher ist es unabdingbar, die Methoden zu optimieren, und ggf. ungenutzte Zeit für die Datensammlung und Bereitstellung zu nutzen, z.B. während des Fluges/Transfers.
Die Sicherheit von 99,99% reicht in diesem Fall nicht aus, da dies bei 4 Millionen Reisenden zu 40000 nicht oder falsch geprüfte Personen führen. Erstrebenswert ist ein Wert nahe an 99,998% da diese Anzahl an Fehlern noch gebilligt und durch zusätzliche Kontrollen verringert und abgearbeitet werden kann. Diese Kontrollen müssen in diesem Fall nicht zusätzlich eingerichtet werden, da manuelle Grenzkontrollen noch lange bestehen bleiben werden, aufgrund der Personengruppen, die für dieses automatisierte biometriegestützte Passwesen nicht geeignet sind, z.B. alte und kranke Menschen oder Kinder.
Um so eine hohe Prüfrate zu erreichen bedarf es schneller aber vor allen Dingen sicherer Systeme. Die Geschwindigkeit spielt beim Einsatz in Bereichen mit hohem Menschenaufkommen einen wesentlichen Faktor. Das System sollte nicht langsamer sein als die bisherigen Kontrollen, denn dann würde sich die Akzeptanz durch den Kunden maßgeblich verringern. Die folgenden Grafiken zeigen kurz, wie die Zeiten ansteigen und die Fehler anwachsen, wenn z.B. die Erkennungsquote bei nur 95% liegt oder die Anzahl der Abfertigungssysteme gering ist. Dabei wurde unterschieden zwischen geringem (10 Passagiere), mittlerem (360 Passagiere) und hohem (800 Passagieren) Personenaufkommen bei Ankunft von nur einem Luftfahrzeug.

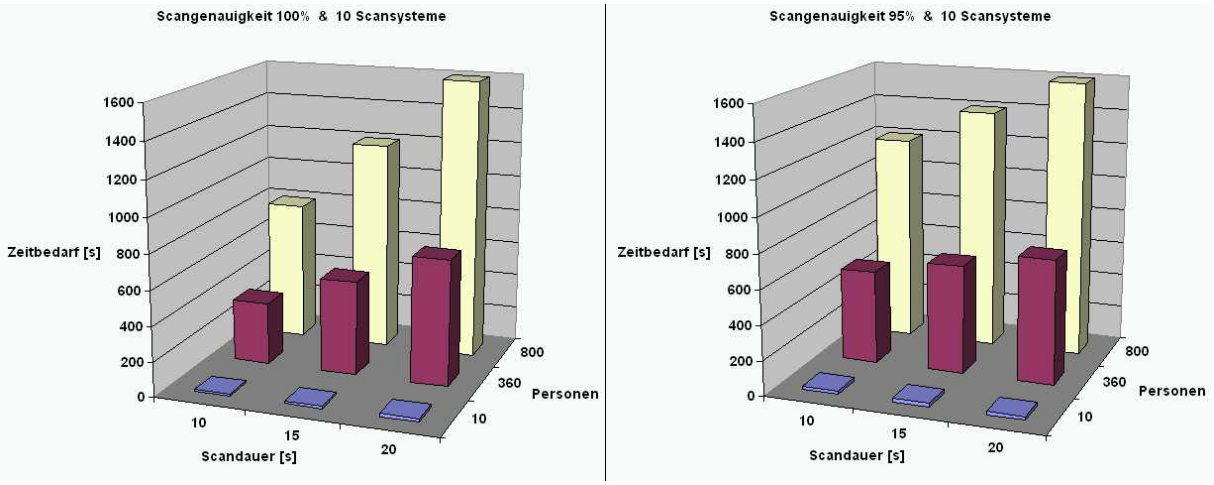


Abbildung 10.4: Scandauerberechnungen

| Scandauer (in s): | 10 | | | 15 | | | 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| Anzahl der Scanvorrichtungen: | 10 | | | | | | | | |
| Anzahl der Passagiere: | 10 | 360 | 800 | 10 | 360 | 800 | 10 | 360 | 800 |
| Scangenauigkeit (in %): | 100% | | | | | | | | |
| Scandauer (in s): | 10 | 360 | 800 | 15 | 540 | 1200 | 20 | 720 | 1600 |
| Fehler: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | 10 | | | 15 | | | 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | | | | | | | | |
| | 10 | 360 | 800 | 10 | 360 | 800 | 10 | 360 | 800 |
| | 100% | | | | | | | | |
| | 10 | 180 | 400 | 10 | 270 | 600 | 10 | 360 | 800 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Scandauer (in s): | 10 | | | 15 | | | 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| Anzahl der Scanvorrichtungen: | 10 | | | | | | | | |
| Anzahl der Passagiere: | 10 | 360 | 800 | 10 | 360 | 800 | 10 | 360 | 800 |
| Scangenauigkeit: | 95% | | | | | | | | |
| Scandauer (in s): | 20 | 540 | 1200 | 20 | 630 | 1400 | 20 | 720 | 1600 |
| Fehler: | 1 | 18 | 40 | 1 | 18 | 40 | 1 | 18 | 40 |

| | 10 | | | 15 | | | 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | | | | | | | | |
| | 10 | 360 | 800 | 10 | 360 | 800 | 10 | 360 | 800 |
| | 95% | | | | | | | | |
| | 20 | 360 | 800 | 20 | 360 | 800 | 20 | 360 | 800 |
| | 1 | 18 | 40 | 1 | 18 | 40 | 1 | 18 | 40 |

Abbildung 10.5: Tabelle zur Scandauerberechnung

Bei der Darstellung der 95% Scangenauigkeit stellen die dargestellten Daten nur die minimalen Zeiten dar. Hinzu kommen noch die Zeiten für die manuelle Kontrolle der Fehlscans. Bei 800 Personen sind das immerhin 40 zusätzliche Prüfungen.
Wie man in der Grafik und Tabelle erkennen kann, spielen zur Bewertung der Leistungsfähigkeit des Systems eine Menge Faktoren eine Rolle. Anzunehmen ist außerdem, dass das durchschnittliche Passagieraufkommen pro Jahr stetig anwachsen wird. Hierbei sind Großraumflugzeuge, mit Sitzkapazitäten von 400 - 800 Sitzen, wie der Airbus A380 oder die Boeing 747, die maßgeblichen Größen. Sollte dann das System nicht effizient genug arbeiten, wird es unweigerlich zu Staus und Behinderung in der Abfertigung kommen. Dies wirkt sich besonders bei dem hohen Personenaufkommen in hochfrequentierten Flughäfen, wie München oder Frankfurt, aus. Bei 30000 - 40000 Passagieren pro Tag in den Hauptreisezeiten, kann ein Stau bei der Abfertigung leicht zu Problemen und Sicherheitsrisiken führen. Dieser Fakt verschlimmert sich dadurch, dass die Grenzkontrollen, in der Regel auf den Flughäfen immer in einem zentralen Bereich durchgeführt werden. Es wird wahrscheinlich nicht jedes Gate mit einem Grenzkontrollbereich ausgestattet sein, sondern nur die jeweiligen Terminals, an denen die Gates angegliedert sind. Da auf den internationalen Airports der Platz für die Grenzkontrollen begrenzt ist, muss ein ausgewogenes Verhältnis zwischen automatisierter und manueller Grenzkontrolle gefunden werden, da sich die beiden Systeme noch lange ergänzen werden. Hierbei stehen sicherlich auch Kosten-/ Nutzenaspekte im Vordergrund, da eine 100%-ige Sicherheit nie erreichbar sein wird, da sie einfach unbezahlbar ist.
Jedoch werden diese Verfahren immer weiter Einzug in den täglichen Reiseverkehr erhalten. Dabei sind die Hauptinitiatoren Europa und die USA, da sich hier derzeit noch der höchste internationale Reiseverkehr abspielt.

## 10.6 Derzeitiger Stand der Einführung in den USA und Europa

### 10.6.1 USA

Die USA sind, seit dem Patriotact das führende Land im Bereich der biometrischen Kontrollen. Alle Fernreisenden bekamen als Auflage für die Ein- und Ausreise maschinenlesbare Reisedokumente zu besitzen, da Ihnen sonst ihre Reisefreiheit eingeschränkt wurde. Somit wurden die Bürger gezwungen, zwangsläufig auf neuere Reisedokumente umzusteigen. Derzeit wird bei jedem Einreisenden der Fingerabdruck gescannt und die persönlichen Daten festgehalten. Zudem müssen längere Aufenthalte mittels Visa beantragt werden, wodurch eine Vorabkontrolle durch die Sicherheitsbehörden möglich sind. Auf diversen Airports werden im Sommer 2005 Kombinationen aus mehreren Methoden getestet:

- **John F. Kennedy Airport (New York und New Jersey)** - RFID-Card und Fingerabdruck-Scanner
- **Logan International Airport (Massachusetts)** - Augen- und Ohrenscan sowie Stimmerkennung bei den GPS[7] gestützten Mobiltelefonen der Sicherheitskräfte
- **Orlando International Airport (Florida)** - Dual-Irisscan

Diese Scans sollen dann später zum Einen für die Identifikation des Flughafenpersonals und zum Anderen auch für die Identifikation der Passagiere verwendet werden. Zudem ist denkbar, dass die gesammelten Daten auch mit den Fahndungsdatenbanken des FBI[8] und der CIA[9] abgeglichen und gespeichert werden, wie es in Europa derzeit schon erprobt wird.
Die biometrischen Verfahren werden in den USA jedoch nicht nur für die Grenzkontrolle und Einreise verwendet, sondern auch zur Identifikation von potentiellen oder schon straffälligen Personen. Hierbei werden von den betroffenen Personen die Fingerabdrücke eingescannt und ein Gesichts-/ oder Irisscan durchgeführt. Diese Daten werden dann den autorisierten Behörden zur Verfügung gestellt, damit diese informiert sind und zudem im Zentralcomputer des FBI gespeichert.

### 10.6.2 Europa & Skandinavien

Europa und Skandinavien sind derzeit mit der Einführung der maschinenlesbaren Reisedokumente beschäftigt und haben gleichzeitig auch eigene Projekte in Bezug auf die Verwendung von biometrischen Daten für den Reiseverkehr in Gang gebracht. So läuft derzeit am Frankfurter Flughafen, für ausgewählte Passagiere und Mitarbeiter der Lufthansa, das „BioP II“-Pilot-Projekt.

[7] Global Positioning System
[8] Federal Bureau of Investigation
[9] Central Intelligence Agency

**BioP II - Flughafen Frankfurt, Deutschland**

Hierbei können sich die Passagiere und Mitarbeiter mittels maschinenlesbarem Ausweisdokument (derzeitiger Reisepass) und Irisscan, identifizieren und die Grenze passieren. Voraussetzung hierfür ist ein, vor der ersten Nutzung, durchgeführtes Enrollment, wobei die persönlichen Daten und die Gültigkeit des Reisepasses geprüft sowie ein Irisscan zur Template-Erstellung durchgeführt wird. Dieses **initiale Enrollment**, in einem gesonderten Datenerfassungsbereich, dauert **ca. 15 - 20 Minuten**!
Die Kontrolle erfolgt im **normalen Betrieb** an separaten Auto-Control-Spuren im Terminal-Bereich des Flughafens [vgl. Abb. 1.6] . Hier muss das Ausweisdokument noch manuell in den Scanner eingelegt werden. Dort wird es gescannt und die ausgelesenen Daten werden mit den Fahndungsdaten von INPOL[10] und SIS[11] abgeglichen. Daran gekoppelt erfolgt ein Irisscan, dessen Daten auch mit den Datenbankensystemen verglichen werden. Das Einlesen und der Abgleich der Daten dauert **ca. 15 - 20 Sekunden**. Bei erfolgloser Identifikation oder bei einer Scan- und Verifikationsdauer von mehr als 20 Sekunden wird man zur manuellen Grenzkontrolle geleitet und das System für den nächsten Nutzer frei gemacht. Derzeit ist dieses Verfahren nur für kleinere Menschenmengen konzipiert und verwendet nur ein biometrisches Verfahren. Dies entspricht noch nicht den von der IATA und ICAO gewünschten Genauigkeiten für das automatisierte Grenzkontrollwesen. Gegebenenfalls muss hier noch ein weiteres Verfahren, z.B. der Fingerabdruckscan hinzugenommen werden, um den Anforderungen zu entsprechen. Dieses Projekt läuft noch bis Mitte August 2005 und wird dann wahrscheinlich in Serie gehen [18].

Abbildung 10.6: Autocontrol-Spur Fraport
Quelle: [18]

**Automatic Boarder Passage - Airport Shiphol, Amsterdam, Niederlande**

Die automatische Grenzpassage ist derzeit für Premium-Kunden am Amsterdamer Flughafen Shiphol mittels Irisscan und Smartcard, der sog. „Privium-Card“, möglich [vgl. Abb

[10]Informationssystem der Polizei
[11]Schengener Informationssystem

1.7] . Hierbei wird Geschäftsleuten und häufig Reisenden dieses spezielle Verfahren zum Preis von ca. 119 Euro / Jahr angeboten, um Ihnen einen schnelleren Transfer zu gewährleisten und bieten zu können. Bei der Grenzkontrolle, wird, wie auch beim BioPII ein Abgleich mit den aktuellen Fahndungsdatenbanken durchgeführt. Dieses Verfahren ist seit Oktober 2002 im Einsatz und hat sich bereits bewährt. Der zunächst verwendete Fingerscan, hat sich als nicht effizient genug herausgestellt und wurde dann durch den Irisscan abgelöst.


Abbildung 10.7: PRIVIUM-Card-Terminal am Shiphol Airport
Quelle: [25]

### 10.6.3 Vergleich

Beide Staatengemeinschaften sind derzeit auf dem besten Wege, die biometrischen Verfahren schnellstmöglich einzuführen und praktikable Standards zu definieren. Durch die stringente Behandlung der Einreisethematik der USA, versucht die EU schnellstmöglich die geforderten Standards bereitzustellen, so dass ein weitestgehend reibungsloser internationaler Reiseverkehr stattfinden kann. In Bezug auf Fortschritt sind beide Staatengemeinschaften gleich auf und unterscheiden sich nur in den Ansätzen zur Realisierung der Vorschläge von der IATA und ICAO. Bemerkenswert ist hierbei, dass weder in den USA noch in der Europäischen Union zwei biometrische Verfahren zum Einsatz kommen, sondern immer nur ein Verfahren in Kombination mit Ausweisdokumenten. Dies zeigt, dass derzeit das Sicherheitsbedürfnis noch nicht so hoch ist und man sich auf die relative Sicherheit eines Systems verlässt. Weiterhin werden die Kontrollen immer noch durch Grenzbeamte überwacht und ggf. unterstützt.
Der stets durchgeführte Abgleich von Personendaten und Fahndungsdatenbanken bedingt, dass die Daten aller Personen gespeichert werden und ggf. für eine Überprüfung durch andere Instanzen zur Verfügung stehen sollen. Diese globale Transparenz der Personendaten hat allerdings auch rechtliche Aspekte und bringt einige Probleme mit sich.

## 10.7 Rechtliche Aspekte und Probleme

Ein sehr wichtiger Aspekt in der Thematik der Biometriedaten stellt die Rechtssprechung dar. Durch die Globalisierung und erweiterte Transparenz aller Bürger in den Staaten, werden oftmals Randgebiete der Rechtsprechung im Bereich des Schutzes persönlicher Rechte berührt und Lücken in diesen Gesetzen, auch von den Regierungen, ausnutzt. Unter dem Deckmantel der Sicherheit und des Terrorschutzes lassen sich viele Rechtslücken problemlos ausnutzen.

### 10.7.1 Datenschutz

Der Datenschutz ist der wesentlichste Teilbereich, der den Bürger direkt und indirekt betrifft. Durch die Zusammenfassung von vielen Daten auf ein Dokument, können diese Daten allesamt ausgelesen und verwendet werden. Hierbei spielt die landesspezifische Feststellung der Datenschutzgesetze eine wichtige Rolle. Hiernach ist es in Deutschland nur den Kontrollbehörden gestattet, die Daten einer Person für genau definierte Zwecke zu nutzen und auszulesen. Das Recht auf Datenschutz ergibt sich derzeit nicht direkt aus dem Grundgesetz, sondern muss aus mehreren Kapiteln abgeleitet werden. Daher fordern Datenschützer die explizite Verankerung im Grundgesetz, damit die Daten einer Person nicht willkürlich verwendet werden dürfen. Jedoch werden derzeit derartige Bestrebungen von der Regierung sehr intensiv geprüft, da man durch eine vorschnelle Gesetzgebung wieder Freiheiten in der Auslegung einbüßen und ungewollte Rechtslücken schaffen würde.
Bei Reisen in die Vereinigten Staaten spielt die deutsche Gesetzgebung nur eine untergeordnete Rolle, da man bei Reisen in ein anderes Land, die dortige Gesetzgebung akzeptieren muss. In den USA werden die Daten zentral beim FBI gespeichert und ggf. verwendet, sobald ein Verdacht besteht. Für EU-Bürger können die Daten problemlos in der gesamten EU verwendet werden, jedoch dürfen diese Daten nur unter Berücksichtigung der EG-Datenschutzrichtlinie (Art. 25 und 26) nicht an Drittländer weitergegeben werden. Für die USA wurde hierzu auch schon eine Ausnahme gemacht, die sog. **„Safe-Harbor-Principles“** regeln den Austausch von Daten nach vorgeschriebenen Richtlinien. Da die USA ein reges Interesse daran haben, möglichst viele Personen erfasst zu haben, ist davon auszugehen, dass bei Einreise in die Vereinigten Staaten, die persönlichen Daten aller Menschen erfasst und auf Dauer gespeichert werden.
Ein weiterer Bereich des Datenschutzes betrifft das Reisedokument an sich. Da die Daten mittels RFID ausgelesen werden können, müssen spezielle Vorrichtungen verwendet werden, damit kein Unberechtigter die biometrischen und sonstigen Daten aus dem Dokument ausliest. Diese Daten sind bei Firmen heiß begehrt, da man damit Kundenprofile erstellen und gezielter abgleichen und auswerten kann. Um dieses unberechtigte Abfragen zu verhindern, rät die IATA das Reisedokument in einer metallischen Schutzhülle zu transportieren, damit der Effekt des Farraday Käfigs ausgenutzt werden kann und somit der Chip und der Besitzer vor Datenraub geschützt ist [12].

### 10.7.2 Kosten

Die Einführung der neuen Verfahren bedingt gleichzeitig die Einrichtung neuer Zertifizierungs- und Identifizierungsstellen. Diese müssen, wie im BioP II- Projekt am Frankfurter Flughafen, mit Scansystemen ausgestattet sein, welche die nötigen Informationen für die biometrischen Datentemplates erzeugen können. Zudem muss, bei Verwendung der RFID-Technik, eine „sichere Instanz“ geschaffen werden, die die Daten auf dem elektronischen Reisepass speichert oder integriert und vor unbefugtem Zugriff geschützt ist.

#### Bürger

Der elektronische Reisepass soll für den Endnutzer in Deutschland 59 Euro kosten. Im Vergleich zu Deutschland soll der amerikanische Pass ca. 50$ und der britische Reisepass ca. 100 Euro kosten.
Dennoch wäre der neue Pass damit doppelt so teuer, wie die bisherigen Exemplare. Zudem konnte bisweilen nicht ausgeschlossen werden, dass zusätzliche Kosten für Identifikation und Speicherung der Daten auf den Nutzer entfallen. Somit könnte man mit einem Gesamtbetrag von ca. 60 - 80 Euro für den elektronischen Reisepass rechnen. Da die Bundesregierung plant, das Reisedokument und die Datenerfassungseinrichtungen im Zeitraum von November 2005 bis Mitte 2007 flächendeckend einführen zu können, werden spätestens 2015 die letzten „alten“ Reisepässe entwertet und ausgetauscht. Somit ist jeder Bürger gezwungen, diesen Reisepass zu erwerben und zu akzeptieren.
Neben den direkten Kosten für den Reisepass werden wahrscheinlich auch verdeckte Kosten wie z.B. erhöhte Flughafensicherheitsgebühren fällig werden, da die Scantechnik auch an den Flughäfen installiert und betrieben werden muss. Somit zahlt der Bürger letztendlich bei jedem Flug für den Komfort und die neu gewonnene Sicherheit des Staates.
Zudem muss der Bürger bei der Passbeantragung ca. 15 - 30 Minuten an Zeit investieren, damit die biometrischen Merkmale aufgenommen werden können.

#### Staat

Die Kosten, für die Einführung der neuen Technik werden derzeit offiziell noch nicht genauer beziffert, da noch einige Parameter unklar sind, ob z.B. jedes Einwohnermeldeamt die Scanvorrichtungen bekommt oder ob zentrale Anlaufstellen in größeren Städten dafür geschaffen werden müssen. Neueste, nicht offizielle Schätzungen belaufen sich auf ca. 120 Millionen Euro pro Jahr und zusätzlich ca. 600 Millionen Euro für die Einführung des neuen Systems [17]. Die Kosten für die Technik und Einrichtungen werden aber sicherlich wieder aus der Staatskasse bezahlt und letztendlich vom Bürger getragen werden.

## 10.8 Schlussbetrachtung und Ausblick

Die Verwendung des automatisierten Pass- und Ausweiswesens in Kombination mit biometrischen Verfahren hat neben vielen Vorteilen auch nicht direkt offensichtliche Nachteile. Die Sicherheit und der Komfort durch die Verwendung von automatisierten Kontrollen muss z.T. teuer auf Kosten des Datenschutzes und Schutz der Persönlichkeit erkauft werden. Dabei ist der schwerwiegendste Faktor im Datenschutz das Data-Mining[12]. Dadurch werden Menschen transparent und viele Schritte vorhersehbar. Zudem entstehen den Kunden und Staaten wesentliche Kosten für die Einführung und den Betrieb von derartigen Systemen. Da die Systeme derzeit noch in der Erprobung und Testphase sind, kann mit einer definitiven Einführung und Nutzung nicht vor 2007 gerechnet werden. Als weiteres Problem muss die beschränkte Einführung der Reisedokumente gesehen werden. Demnach ist damit zu rechnen, dass ein erheblicher Anteil an Reisenden die manuelle Grenzkontrolle in Anspruch nehmen muss, da die neue Technik nicht in allen Ländern der Erde eingeführt wird und auch nicht auf alle Menschen angewendet werden kann, da sich die biometrischen Merkmale zu stark ändern oder nicht effektiv ausgelesen werden können. Hierbei richtet sich das Augenmerk vornehmlich auf Kinder, leistungsgeminderte Personen oder Reisende aus Staaten ohne den technologischen Fortschritt. Bei vielen Staaten der dritten Welt, werden derartige Dokumente wahrscheinlich noch lange auf sich warten lassen.
Als Vorteile lassen sich die extrem einfache Handhabung und die hohe Zuverlässigkeit der derzeit verwendeten Systeme anbringen, die ein komfortables Reisen ermöglichen. Weiterhin können durch die Anwendung dieser neuen Technik im Visa-Verfahren, Aufwände für Personal und Bürokratie minimiert und das Verfahren zur Erteilung von Visa vereinfacht werden. Die Identifikation mittels Biometrie ermöglicht es zudem den Staatsorganen, potentiell gefährliche Personen schnell zu identifizieren und gesondert prüfen und behandeln zu können. Die Gesellschaft könnte so vor wiederholt straffällig gewordenen Personen geschützt werden. Wobei dieser Aspekt jedoch mit Vorsicht zu genießen ist, da die Personen wahrscheinlich ein Leben lang als Straftäter deklariert werden, auch wenn die Schuld schon längst verbüßt wurde.
In naher Zukunft werden sicherlich die derzeitigen Verfahren verfeinert und das automatisierte biometriegestützte Kontrollwesen auch in vielen anderen Bereichen Einzug erhalten. Zudem könnten bald auch genetische Merkmale auf dem elektronischen Ausweis Einzug halten. Hierbei ist vor allen Dingen der genetische Fingerabdruck zu nennen. Dies würde eine nahezu 100%-ige Identifikation ermöglichen. Allerdings ist die Technik derzeit zu langsam und aufwändig, als dass man sie für Schnelltests verwenden könnte. Dadurch scheidet diese Technik zunächst in hochfrequentierten Bereichen aus. Letztendlich muss man sich darüber im Klaren sein, wie weit die Kontrollen gehen dürfen, bevor sie die Freiheit zu sehr einschränken. Es heißt zwar: **“Ewige Wachsamkeit ist der Preis der Freiheit“**, aber es sollte auf jeden Fall verhindert werden, dass die totale Kontrolle aller Bürger, wie in George Orwells Neuzeit-Utopie „1984“, Realität wird.

**BIG BROTHER IS WATCHING YOU**

[12]Sammeln von Daten der Menschen

# Literaturverzeichnis


[1] Ian Williams, www.idsysgroup.com, biometrics 101 ISG.pdf, 2003

[2] Technical Report, *Biometrics deployment of Machine Readable Travel Documents 2004*, www.icao.com, Biometrics deployment of Machine Readable Travel Documents 2004.pdf, 2004

[3] Tom A.F. Kinneging, PKI Task Force,*TR-PKI mrtds ICC read-only access v1 1*, www.icao.com, TR-PKI mrtds ICC read-only access v1 1.pdf, 2004

[4] Technical Study / Report, *Annex A - Photograph Guidelines*, www.iata.com, Annex A - Photograph Guidelines.pdf, 2004

[5] Technical Study / Report, *Annex B - Facial Image Size Study 1*, www.iata.com, Annex B - Facial Image Size Study 1.pdf, 2004

[6] Technical Study / Report, *Annex C - Facial Image Size Study 2*, www.iata.com, Annex C - Facial Image Size Study 2.pdf, 2004

[7] Technical Study / Report, *Annex D - Face Image Data Interchange*, www.iata.com, Annex D - Face Image Data Interchange.pdf, 2004

[8] Technical Study / Report, *Annex E - Iris Image*, www.iata.com, Annex E - Iris Image.pdf, 2004

[9] Technical Study / Report, *Annex F - Fingerprint Image*, www.iata.com, Annex F - Fingerprint Image.pdf, 2004

[10] Technical Study / Report, *Annex G - Fingerprint Minutiae*, www.iata.com, Annex G - Fingerprint Minutiae.pdf, 2004

[11] Technical Study / Report, *Annex H - Fingerprint Pattern*, www.iata.com, Annex H - Fingerprint Pattern.pdf, 2004

[12] Technical Study / Report, *Annex I - Contactless ICs*, www.iata.com, Annex I - Contactless ICs.pdf, 2004

[13] Technical Study / Report, *Annex K - ICAO Supplementary Requirements to ISO14443-v2*, www.iata.com, Annex K - ICAO Supplementary Requirements to ISO14443-v2.pdf, 2004

[14] Technical Study / Report, *Annex L - ePassports Data Retrieval Test Protocol*, www.iata.com, Annex L - ePassports Data Retrieval Test Protocol.pdf, 2004

[15] BSI,*Digitale Sicherheitsmerkmale im E Pass*, www.bis.bund.de, Sicherheitsmerkmale epass.pdf, 2005

[16] Christiane Schulzki-Haddouti, *Biometrie ohne Nebenwirkungen?*, Zeitschrift: c't - magazin für computer technik, Ausgabe 10/05 S. 94-95, 2005

[17] Richard Sietmann, *Der Biometrie-Pass kommt,* Zeitschrift: c't - magazin für computer technik, Ausgabe 13/05, S.44-45, 2005

[18] www.bgs.de, *Automatisierte biometrische Grenzkontrolle(ABG)*, 2005

[19] Web-Seite Frankfurter Flughafen, www.fraport.de, 2005

[20] *Gesetze des Patriot-Act im Internet*, www.patriotact.com, 2005

[21] Dr. Helmut Bäumler, Lukas Gundermann, Dr. Thomas Probst, *Stand der nat. und internat. Diskussion zum Thema Datenschutz bei biometrischen Systemen*, Landeszentrum für Datenschutz Kiel, tabga.pdf, 2001

[22] J.Scott Bechtel et al., *Purdue University Innovation Realization Lab White Paper*, www.pen-one.com, Pen-One development white paper.pdf, 2004

[23] WDR-Reportage, *130 Euro für einen neuen Reisepass*, http://www.wdr.de/themen/politik/deutschland/biometrie pass/050113.jhtml, 2003

[24] WDR-Quarks und Co, *Skript zur WDR-Sendereihe Quarks & Co - Big Brother is Watching*, www.quarks.de/pdf/Q big brother.pdf, 2004

[25] Airport Technology, *Amsterdam Airport Schiphol Expansion*, http://www.airport-technology.com/projects/schiphol/schiphol6.html, 2005

[26] Büro für Techikfolgen-Abschätzungen beim deutschen Bundestag, *Biometrie und Ausweisdokumente, 2. Sachstandsbericht*, www.tab.fzk.de, ab93.pdf, 2003

[27] Bori Toth, *Suitability of Biometrics at Airports*, www.deloitte.co.uk/biomerics, BiometricsAtAirports Moscow2005.pdf, 2005