Andreas Dewald, Felix C. Freiling (Hrsg.)

Forensische Informatik

2. Auflage

Dr.-Ing. Andreas Dewald
Prof. Dr.-Ing. Felix C. Freiling
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department Informatik
Martensstr. 3
91058 Erlangen
https://www1.informatik.uni-erlangen.de

Weitere Autoren:

Michael Gruhn
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department Informatik
Martensstr. 3
91058 Erlangen

Dr.-Ing. Christian Riess
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department Informatik
Martensstr. 3
91058 Erlangen

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

1. Klassische forensische Wissenschaften
    - 1.1 Forensische Wissenschaften
        - 1.1.1 Die wissenschaftliche Methode
        - 1.1.2 Zur Dauerhaftigkeit wissenschaftlicher Erkenntnisse
        - 1.1.3 Forensische Wissenschaften, Kriminalistik und Kriminologie
        - 1.1.4 Forensische Wissenschaft und das Rechtssystem
        - 1.1.5 Frühe literarische Bezüge
        - 1.1.6 Analyse von Spuren zur Identifizierung von Personen
    - 1.2 Spuren und ihre Entstehung
        - 1.2.1 Spureninformation und Spurenträger
        - 1.2.2 Integrität und Authentizität von Spuren
        - 1.2.3 Klassifikation von Spuren
        - 1.2.4 Übertragung (Transfer)
        - 1.2.5 Zerteilbarkeit (Divisibility)
        - 1.2.6 Übertragung von Mustern
        - 1.2.7 Theorie des Transfers
    - 1.3 Rekonstruktion eines Tathergangs
        - 1.3.1 Ereignisse
        - 1.3.2 Assoziation
        - 1.3.3 Der Weg zur Assoziation
        - 1.3.4 Beispiele
        - 1.3.5 Notwendigkeit von Identifizierung und Klassifizierung
    - 1.4 Zusammenfassung

2. Digitale Spuren
    - 2.1 Definition und Abgrenzung

2.1.1 Information und Träger bei digitalen Spuren
2.1.2 Integrität
2.1.3 Authentizität
2.2 Entstehung digitaler Spuren
2.2.1 Zerteilbarkeit und Transfer in der digitalen Welt
2.2.2 Geschlossenes vs. offenes System
2.2.3 Abstraktionsschichten
2.3 Eigenschaften digitaler Spuren
2.3.1 Flüchtigkeit
2.3.2 Technische Vermeidbarkeit
2.3.3 Manipulierbarkeit
2.3.4 Kopierbarkeit
2.3.5 Semantik
2.4 Assoziation mittels digitaler Spuren
2.4.1 Identifizierung
2.4.2 Klassifizierung
2.4.3 Individualisierung
2.4.4 Assoziation
2.4.5 Beispiel: Multimediaforensik
2.4.6 Beispiel: USB-Speichergeräte
2.4.7 Beispiel: Browser Cache
2.4.8 Beispiel: Copy/ Move Operationen im Hauptspeicher
2.4.9 Quantifizierung der Irrtumswahrscheinlichkeit
2.5 Assoziation als zentraler Aspekt der forensischen Informatik
2.5.1 Computerforensik
2.5.2 Forensische Informatik
2.5.3 Fragestellungen der forensischen Informatik im engeren Sinne
2.5.4 Fragestellungen der forensischen Informatik im weiteren Sinne
2.6 Verwandte Literatur
2.6.1 Zur Integration der Informatik in die Reihe forensischer Wissenschaften
2.6.2 Zur Praxis der Computerforensik
2.7 Zusammenfassung

3. Spuren, Interferenz und die Lösbarkeit von Rekonstruktionsproblemen

- 3.1 Intuition
- 3.2 Modell eines digitalen Systems
  - 3.2.1 Variablen
  - 3.2.2 Programme
  - 3.2.3 Pfade
  - 3.2.4 Nichtdeterminismus
  - 3.2.5 Aktionen und Operationen
  - 3.2.6 Weitere Annahmen
- 3.3 Problemdefinition
  - 3.3.1 Allgemeines Rekonstruktionsproblem
  - 3.3.2 Spezifisches Rekonstruktionsproblem
  - 3.3.3 Spezifisches Gruppen-Rekonstruktionsproblem
- 3.4 Spuren
  - 3.4.1 Definition
  - 3.4.2 Charakteristische Spuren
  - 3.4.3 Gemeinsame charakteristische Spuren
- 3.5 Kontraspuren
  - 3.5.1 Definition
  - 3.5.2 Charakteristische Kontraspuren
  - 3.5.3 Gemeinsame charakteristische Kontraspuren
- 3.6 Interferenz
  - 3.6.1 Allgemeine Interferenz
  - 3.6.2 Variablen-Interferenz
  - 3.6.3 Spuren-Interferenz
  - 3.6.4 Kombinationen von Interferenz
  - 3.6.5 Interferierende Aktionen, Überdeckung und Äquivalenz
- 3.7 Rekonstruktion
  - 3.7.1 Das zugrundeliegende System
  - 3.7.2 Lösbarkeit des SRP
  - 3.7.3 Lösbarkeit des SGRP
  - 3.7.4 Lösbarkeit des GRP
- 3.8 Zusammenfassung

4. Einführung in die Multimediaforensik
   - 4.1 Themen der Multimediasicherheit
     - 4.1.1 Steganographie: Verdeckte Kommunikation
     - 4.1.2 Watermarking: Schutz geistigen Eigentums

4.1.3 Blinde Bildforensik: Authentizität eines Bildes oder Videos
4.2 Identifikation des Aufnahmegeräts
4.2.1 Identifikation des Kameramodells
4.2.2 Sensorrauschen zur Identifikation der Kamera
4.3 Inhaltliche Erkennung von Bildmanipulationen
4.3.1 Spuren in Kompressionsartefakten
4.3.2 Erkennung von Copy-Paste-Fälschungen
4.3.3 Erkennung von Skalierung und Rotation
4.3.4 Szenenkonsistenz
4.4 Werkzeuge für Forschung und Entwicklung
4.4.1 Prototyping
4.4.2 Applikationsentwicklung

5. Vorgehensmodelle
5.1 Hypothesenbasiertes Arbeiten
5.2 Das ***Incident-Response***-Modell
5.2.1 Incident Response
5.2.2 Das Vorgehensmodell
5.3 Der investigative Prozess
5.3.1 Anschuldigung
5.3.2 Güterabwägung
5.3.3 Tatortsicherung
5.3.4 Beschlagnahme
5.3.5 Sicherung
5.3.6 Bergung
5.3.7 Auswertung
5.3.8 Reduktion
5.3.9 Analyse
5.3.10 Bericht
5.3.11 Bezeugen
5.4 Das Common Model
5.4.1 Überblick
5.4.2 Pre-incident Preparation
5.4.3 Pre-Analysis Phase
5.4.4 Analysis Phase
5.4.5 Post-Analysis Phase
5.4.6 Diskussion

5.5 Weitere Modelle
5.6 Zusammenfassung

6. Die Methodik der forensischen Informatik am Beispiel Partitionssysteme
    6.1 Datenträgertechnologien
        6.1.1 Festplatten
        6.1.2 Flash-Speicher
        6.1.3 Schnittstellen
        6.1.4 Host Protected Area (HPA)
        6.1.5 Device Configuration Overlay (DCO)
        6.1.6 Zugriff auf den Datenträger als Teil des Boot-Prozesses
    6.2 Partitionssysteme und ihre Analyse
        6.2.1 Terminologie
        6.2.2 Vor- und Nachteile von Partitionen
        6.2.3 Adressierungsarten
        6.2.4 Partitionstabelle
        6.2.5 Analyse ohne Partitionsinformationen
        6.2.6 Ausblick auf konkrete Partitionssysteme
    6.3 DOS/MBR
        6.3.1 Primäre und Sekundäre Partitionen
        6.3.2 Sonderfälle
    6.4 Globally Unique Identifier (GUID) Partition Table (GPT)
        6.4.1 Protective MBR (PMBR)
        6.4.2 GPT Header
        6.4.3 GPT Partitionseintrag
    6.5 Forensischer Zugriff auf Datenträgerdaten
        6.5.1 Arten des Zugriffs
        6.5.2 Schutz der Originaldaten
        6.5.3 Fehlerbehandlung
        6.5.4 Abstraktionsstufe und Granularität des Zugriffs
        6.5.5 Wahrung der Integrität
    6.6 Zusammenfassung

7. Dokumentation
    7.1 Allgemeine Aspekte
        7.1.1 Verwahrungskette (engl. ***chain of custody***)
        7.1.2 Handschriftliche Dokumentation

7.1.3 Automatische Dokumentation
7.1.4 Dokumentation von Zeit
7.1.5 Vorgeschriebene Mindestanforderungen
7.2 Nachvollziehbarkeit
7.2.1 Nachvollziehbarkeit vs. Nachprüfbarkeit
7.2.2 Verhältnismäßigkeit der Dokumentation
7.2.3 Versionierung
7.2.4 Beispiele
7.3 Aufbau forensischer Berichte
7.3.1 Zielpublikum
7.3.2 Grobgliederung
7.4 Vorgehen bei der Erstellung
7.5 Beispiele aus forensischen Berichten
7.5.1 Bericht 1
7.5.2 Bericht 2
7.5.3 Bericht 3
7.5.4 Bericht 4
7.5.5 Bericht 5
7.6 Zusammenfassung

8. Praktische Aspekte digitaler Ermittlungen
8.1 Organisatorische Aspekte
8.1.1 Rollen und Aufgabenverteilung
8.1.2 Komplexe Durchsuchungen
8.2 Priorisierung und Auswahl bei der Sicherung
8.2.1 Grundsätzliche Erwägungen
8.2.2 Technische Rahmenbedingungen
8.2.3 Aktuelle Praxis bei der Datensicherung
8.2.4 Drei grundsätzliche Sicherungsmethoden im Überblick
8.2.5 Sicherung auf der physischen Ebene
8.2.6 Sicherung auf der Partitionsebene
8.2.7 Sicherung auf der Dateiebene
8.2.8 Sicherung als Teil der Live-Analyse
8.2.9 Beweiswert und Verhältnismäßigkeit
8.3 Organisation und Analyse großer Datenmengen
8.3.1 Priorisierung der Daten
8.3.2 Aufbereitung von Massendaten

8.4 Zusammenfassung

Literatur

Index

Abbildungsverzeichnis

1.1 Filter zur Sammlung mikroskopischer Spuren
1.2 Schematische Darstellung von Locards Austauschprinzip
1.3 Transfer von Texturen auf Bruchstücke eines Objekts
1.4 Übereinstimmende Bruchkanten in einem Kabel
1.5 Kratzspuren eines Werkzeugs (oben) auf einem verformbaren Material
1.6 Schematische Darstellung des Laufs einer Schusswaffe
1.7 Schema eines Vergleichsmikroskops
1.8 Vergleich von Spuren auf Kugeln
1.9 Beispiel für gleichzeitigen Austausch von Muster und Materie
1.10 Schematische Darstellung des Weges zur Assoziation

2.1 Die Erweiterung von Locards Austauschprinzip auf den digitalen Tatort nach Casey
2.2 Zustand des 3-Bit Automaten vor und nach der Ausführung eines mov Befehls

3.1 Beispielhafte Visualisierung eines Zustandsautomaten
3.2 Beispielhafte Visualisierung des zu rekonstruierenden Plades in einem Zustandsautomaten
3.3 Beispielhafte Visualisierung alternativer Pfade in einem Zustandsautomaten
3.4 ***Programm 1.***
3.5 Pfade von ***Programm 1.***
3.6 Zustandsübergangsdiagramm von ***Programm 1.***
3.7 Visualisierung der drei orthogonalen Kategorien des spezifischen Rekonstruktionsproblems
3.8 ***Programm 2.***
3.9 Zustandsübergangsdiagramm von ***Programm 2.***
3.10 ***Programm 3.***
3.11 Zustandsübergangsdiagramm von ***Programm 3.***
3.12 ***Programm 4.***

3.13 Zustandsübergangsdiagramm von ***Programm 4.***
3.14 ***Programm 5.***
3.15 Zustandsübergangsdiagramm von ***Programm 5.***
3.16 ***Programm 6.***
3.17 Zustandsübergangsdiagramm von ***Programm 6.***
3.18 Schematische Darstellung zweier Mengen bei ***Nicht-Interferenz***
3.19 Schematische Darstellung zweier Mengen ***M*** und ***N*** bei ***schwacher Interferenz***
3.20 Schematische Darstellung zweier Mengen mit ***starker Interferenz***
3.21 Schematische Darstellung zweier Mengen ***M*** und ***N*** bei ***absoluter Interferenz***
3.22 Schematische Darstellung der Variablenmengen bei ***starker Variablen-Interferenz***
3.23 Schematische Darstellung der Variablenmengen bei ***absoluter Variablen-Interferenz***
3.24 ***Programm 7.***
3.25 Zustandsübergangsdiagramm von ***Programm 7.***
3.26 Schematische Darstellung einer nicht minimalen Überdeckung der Variablenmenge von Aktion ***σ***
3.27 Schematische Darstellung einer nicht minimalen Überdeckung der Spurenmenge von Aktion ***σ***
3.28 ***Programm 8.***
3.29 Zustandsübergangsdiagramm von ***Programm 8.***
3.30 Rekonstruiertes Zustandsübergangsdiagramm von ***Programm 8.***
3.31 ***Programm 9.***
3.32 Zustandsübergangsdiagramm von ***Programm 9.***

4.1 Steganographie: Prisoner’s Model
4.2 Einbettung eines digitalen Wasserzeichens
4.3 Auslesen eines digitalen Wasserzeichens
4.4 Ortsraum und Frequenzraum
4.5 Wasserzeicheneinbettung im Frequenzraum
4.6 Ansatzpunkte für bildforensische Methoden
4.7 Lichtbrechung unter lateraler chromatischer Aberration
4.8 Anschauungsbeispiel für laterale chromatische Aberration
4.9 Illustration von JPEG-Kompressionsartefakten
4.10 Beispiel für Copy-Paste-Manipulation

4.11 Beispiel zur Erkennung von Skalierung
4.12 Oberflächennormalen in einer Szene

5.1 Breite vs. Tiefe einer hypothesenbasierten, forensischen Untersuchung.
.
5.2 The Incident Response Process
5.3 Der investigative Prozess
5.4 Der investigative Prozess in der Übersetzung von Dornseif
5.5 Arten digitaler Geräte am Tatort
5.6 Unterschiedliche Arten von Kabeln
5.7 Beschreibung von Speichermedien
5.8 Sonstige elektronische Spuren
5.9 Beispiel für die Dokumentation der vorgefundenen Verkabelung
5.10 Überblick über das Common Model
5.11 Common Model: Pre-Analysis Phase
5.12 Common Model: Analysis Phase
5.13 Common Model: Post-Analysis Phase

6.1 Bild eines IDE-Anschlusskabels
6.2 Bild eines S-ATA Datenkabels
6.3 Illustration zur CHS-Adressierung
6.4 Mögliche Partitionsanordnungen
6.5 Master Boot Record
6.6 CHS-Adressierung wie in MBR
6.7 GPT Layout mit PMBR
6.8 GPT Layout-Beispiel

7.1 Untersuchter Rechner und Untersuchungsrechner
7.2 Grobgliederung eines Untersuchungsberichts
7.3 Gliederung von Bericht 1
7.4 Inhaltsverzeichnis von Bericht 2 (erster Teil)
7.5 Inhaltsverzeichnis von Bericht 2 (zweiter Teil)
7.6 Auszug aus Bericht 5

8.1 Hierarchische Gliederung von Speichermedien
8.2 Schematische Darstellung der sechs Schritte zur Aufbereitung von Daten

Einleitung

„Cyberkriminalität“ bezeichnet in der Regel diejenige Kriminalität, die sich auf vernetzte Computertechnologie bezieht. Gemeint sein können einerseits klassische Delikte, die in den Cyberspace übertragen werden, also beispielsweise Betrug, Beleidigungen oder Urheberrechtsverletzungen. Andererseits evoziert die Existenz von Computertechnologie aber auch gänzlich neue Deliktsformen, bei denen Computer das Angriffsziel darstellen. Zu diesen neuen Formen von Kriminalität zählen etwa Computersabotage, Industriespionage und der Diebstahl persönlicher Daten.

Dass Cyberkriminalität heute durch die Medien und durch die Politik eine große Bedeutung zugemessen wird, hat mehrere Gründe: Zum einen ist das Thema gerade innerhalb medial bevorzugter Zielgruppen so nachrichtentauglich, dass sich ein einzelner großer Sicherheitsvorfall schnell zu einer die Medienlandschaft beherrschenden Schlagzeile entwickelt. Zum anderen haben die IT-Sicherheitsindustrie und Teile der Sicherheitsbehörden beim Wettbewerb um Stellen und Ressourcen oder bei der Sicherung einer Monopolstellung am Markt ein spezielles Eigeninteresse daran, dass sich die Öffentlichkeit wegen einer verstärkten Bedrohungslage im Cyberspace um ihre Sicherheit sorgt. Und dass, obwohl bislang so gut wie keine repräsentativen Studien zu den tatsächlichen Schäden von Cyberkriminalität vorliegen (Brodowski u. Freiling, 2011).

Als gesichert gilt hingegen allein aufgrund der zunehmenden Durchdringung unserer Gesellschaft mit Computertechnologie, dass der Umfang von Cyberkriminalität in den letzten Jahren zugenommen hat und weiter zunehmen wird. Dieser Umstand führt zwangsläufig zu einer erhöhten Relevanz von Computern in straf- oder zivilrechtlichen Verfahren vor Gericht. Deshalb steigt auch der Bedarf an Spezialisten, die derartige Beweismittel „gerichtsfest“ analysieren und ihre Ergebnisse auch für Juristen verständlich darstellen können. In der Folge ist ein pragmatisches technisches Sachverständigenwesen entstanden, das in Deutschland unter dem Oberbegriff „Computerforensik“ oder „digitale Forensik“ firmiert (Geschonneck, 2006; Casey, 2004).

Wie in anderen Ländern auch, entstammt die Computerforensik praktischen Bedürfnissen: Computer sind seit langem derart komplex, dass ihre Funktionsweise mit Blick auf das jeweilige Verfahren vor Gericht nur noch von technischen Experten, meist Informatikern, erklärt werden kann. Diese Computerexperten genossen einst geradezu blindes Vertrauen, schätzte man sich doch glücklich, überhaupt jemanden gefunden zu haben, der sich mit einer spezifischen Technologie auskannte. Bereits 2003 stellten Rogers u. Seigfried (2003, S. 1) fest,

> ***„[...] that there is a disproportional focus on the applied aspects of computer forensics, at the expense of the development of fundamental theories.“***

Heute setzt sich verstärkt die Einsicht durch, dass auch bei der Analyse von Computern wissenschaftliche Methoden angewendet werden müssen. Denn allein wissenschaftlich anerkannte Methoden garantieren die größtmögliche Objektivität der Analyseergebnisse.

Der Trend zu wissenschaftlicher Methodik bei der Analyse von Beweismitteln vor Gericht ist nicht neu. Viele Wissenschaften haben sich bereits mehr oder weniger prominent in den Dienst der Rechtsgelehrten gestellt, beispielsweise die Medizin, die Physik oder die Biologie. Wegen ihres Bezugs zum Rechtssystem bezeichnet man die entsprechenden Bereiche als ***forensische*** Wissenschaften. In diesem Kontext schien die Informatik bisher eine Sonderrolle zu spielen, ging es hier doch um eine vermeintlich neue Art von Spuren: Denn während es früher im Wesentlichen ***physische Spuren*** waren, die man mit wissenschaftlichen Methoden untersuchte (***physical evidence***), stehen bei der digitalen Forensik ***digitale Spuren*** (***digital evidence***) im Mittelpunkt des Interesses.

Zwar entstehen aus der Natur der digitalen Spuren einige Besonderheiten. Trotzdem soll dieser Text belegen, dass die forensische Informatik eine „ganz normale“ forensische Wissenschaft ist und kein neuartiger „Hokuspokus“. Die Autoren möchten mit diesem Text fördern, dass sich die Informatik etwas deutlicher als bisher in die Reihe der forensischen Wissenschaften stellt. Mit diesem Postulat geht die Vorstellung einher, dass sich die pragmatische Computerforensik zu einer forensischen Wissenschaft entwickelt, eben einer ***forensischen Informatik***.

Viele Methoden der Informatik können auf Fragen des Rechtssystems angewendet werden. Beispiele sind die Bereiche Datenbanken, maschinelles

Lernen, Softwaretechnik, Bildverarbeitung und Betriebssysteme. Das Gebiet der forensischen Informatik ist demzufolge sehr umfassend. Ziel dieses Textes ist es nicht, jeden dieser Bereiche erschöpfend darzustellen. Hauptsächliche Absicht der Autoren ist vor allem, zur Strukturierung und zur methodischen Fundierung des Gebietes beizutragen, und in einigen ausgewählten Bereichen einige für das Gebiet relevante Fragen in der Tiefe zu diskutieren.

Die vorliegenden Ausführungen richten sich an interessierte Personen, die sich im Bereich der forensischen Informatik mit fundiertem Hintergrundwissen ausstatten wollen. Die Autoren setzen voraus, dass ihre Leser über ein grundlegendes technisches Verständnis von Hard- und Software verfügen. Allerdings sind viele Teile der Darstellung auch ohne diese Grundlagen verständlich, also allein aus der Anwenderperspektive heraus. Zudem versucht vorliegender Text, in Ergänzung zu den hervorragenden englischsprachigen Basiswerken wie denen von Casey (2004) oder Farmer u. Venema (2005) eine spezifisch deutsche Perspektive einzunehmen sowie ein theoretisches Fundament für das pragmatische Einführungswerk von Geschonneck (2006) zu bieten. Der Fokus liegt auf Einsichten mit relaltiv langer Halbwertszeit. Wir müssen also leider auf detaillierte Instruktionen für die Auswertung konkreter digitaler Spuren größtenteils verzichten.

In die Darstellung sind die Erfahrungen aus mehreren Jahren Forschung und Lehre im Bereich der forensischen Informatik eingeflossen. Die erste universitäre Lehrveranstaltung in Deutschland zu diesem Thema entstand im Sommersemester 2005 unter Federführung von Maximillian Dornseif an der RWTH Aachen. Seitdem haben die Autoren diese Lehrveranstaltung jährlich angeboten und kontinuierlich weiterentwickelt, zunächst an der Universität Mannheim und seit 2011 an der Friedrich-Alexander-Universität Erlangen-Nürnberg. Die Weiterentwicklung der Erkenntnis spiegelt sich in der Namensgebung wider: Hieß die Aachener Vorlesung noch „ Computerforensik“ so wurde sie in Mannheim zunächst in „digitale Forensik“ umbenannt. Mittlerweile heißt die Vorlesung „Forensische Informatik“.

Weitere Erfahrungen konnten die Autoren bei der Gestaltung des ersten berufsbegleitenden Masterstudiengangs zum Thema „digitale Forensik“ sammeln, der von der Hochschule Albstadt-Sigmaringen in Fernlehre angeboten wird. Folgender Text kann durchaus als Skriptum für eine einführende Lehrveranstaltung in den Bereich der forensischen Informatik verwendet werden. Er sollte aber um konkrete technische Abschnitte erweitert werden, etwa

hinsichtlich Datenträgeruntersuchungen durch das hervorragende Buch von Carrier (2005).

Ausblick auf das Buch

Der Text ist in einzelne Kapitel gegliedert, die von unterschiedlichen Autorenkollektiven stammen. Kapitel 1 gibt einen Überblick über die klassischen forensischen Wissenschaften, die sich im Wesentlichen mit ***physischen*** Spuren beschäftigen. In Kapitel 2 folgt der Übergang zu digitalen Spuren. Dort wird die Informatik als forensische Wissenschaft definiert. In Kapitel 3 werden die Konzepte und Fragestellungen der forensischen Informatik formalisiert und in eine allgemeine Theorie der Rekonstruktion diskreter Zustandsfolgen eingebettet.

Die weiteren Kapitel übertragen wichtige Aspekte der klassischen Forensik in den Bereich der Informatik. Kapitel 4 gibt eine Einführung in die Untersuchung von Multimediadaten. Kapitel 5 diskutiert Regeln und Modelle, mit denen das Vorgehen bei der Untersuchung digitaler Spuren strukturiert werden kann. Kapitel 6 führt forensische Fragestellungen im Bereich der Datenträgeranalyse am Beispiel von Partitionssystemen aus. Kapitel 7 behandelt den wichtigen Aspekt der Dokumentation. Abschließend bietet Kapitel 8 einen Überblick über verschiedene praktisch relevante Aspekte einer digitalen Ermittlung wie etwa den Umgang mit großen Datenmengen.

Zur Entstehung dieses Buches

Die erste Auflage dieses Buches, die im Oktober 2011 erschien, verdanken wir dem Projekt „Master Online Digitale Forensik“ (Brodowski u.a., 2014), das zwischen 2009 und 2012 durch die Landesstiftung Baden-Württemberg gefördert wurde. Der Text wurde in der Folge mehrfach stark überarbeitet und erweitert.

Um der Vielfalt und Breite des Gebietes Rechnung zu tragen, sind wir in der zweiten Auflage des Buches von der Autoren- in die Herausgeberrolle gewechselt und haben das Buch auch für Beiträge anderer Autoren geöffnet. Zu

nennen ist hier insbesondere der Beitrag von Christian Riess in Kapitel 4. Neu ist auch Kapitel 6 über Partitionssysteme, das im Rahmen des durch das BMBF und den Europäischen Sozialfonds geförderten Projekts „Open C3S“ entstand.

Danksagungen

Der rege Kontakt zu forensischen Praktikern aus Strafverfolgung und Privatwirtschaft bildet eine der zentralen Triebfedern unserer Arbeit. Wir möchten uns sehr für die offene Kommunikation und die nützlichen Rückmeldungen zu unserer Arbeit bedanken.

Für weitere hilfreiche Unterstützung bei der Erstellung dieses Textes danken wir außerdem Harald Baier, Georg Blome, Rainer Böhme, Sabine Braak, Jana Dittmann, Marco Drummer, Hans-Georg Eßer, Ulrike Freiling, Andreas Grimm, Barbara Haeberlin, Daniel Hammer, Stefan Handel, Marc Junginger, Robert Kandzia, Stefan Kiltz, Benjamin Klink, Christoph Klöcker, Marion Liegl, Benedikt Lorch, Christoph Mattejat, Sebastian Nemetz, Mark Pollitt, Michael Portner, Romy Rahnfeld, Martin Rieger, Konstantin Sack, Thomas Salzberger, Peter Scholz, Marco Schuba, Jörg Schwenk, Daniel Selényi, Marco Siegert, Hans Spath, Robert Spielmann, Jürgen Straub, Michael Tielemann, Victor Völzow, Bärbel Wolf-Gellatly, Cengizhan Yücel, sowie allen Studierenden, mit denen wir in der Vergangenheit zusammengearbeitet haben und unseren Familien und Melanie.

Rückmeldungen

Trotz zahlreicher Verbesserungsvorschläge vorgenannter Personen enthält dieser Text sicherlich noch Fehler, seien sie inhaltlich, konzeptionell, sprachlich oder stilistisch. Die Herausgeber freuen sich deshalb über jede konstruktive Rückmeldung, die wir in zukünftigen Versionen unseres Textes gerne berücksichtigen.

Andreas Dewald
Felix Freiling

Kapitel 1

Klassische forensische Wissenschaften

Autoren: Felix Freiling, Andreas Dewald

Bevor wir uns mit digitaler Forensik als solcher beschäftigen können, betrachten wir in diesem Kapitel zunächst Forensik im Allgemeinen, denn die Entwicklung forensischer Wissenschaften begann lange vor der Erfindung des digitalen Computers. Auch wenn Methoden der digitalen Forensik mittlerweile Eingang in den Kanon der allgemein akzeptierten forensischen Wissenschaften gefunden haben, sprechen wir manchmal von der ***klassischen*** Forensik, wenn wir explizit die forensischen Wissenschaften vor der Einführung des Computers meinen, also die ***nicht-digitale*** Forensik. Bezeichnungen wie ***analoge*** Forensik (als Gegenpol zu ***digital***) oder ***virtuelle*** Forensik bzw. ***Cyberforensik*** (als Gegenpol zu ***physisch***) führen unserer Erfahrung nach in die Irre und werden nicht verwendet.

1.1 Forensische Wissenschaften

Das Attribut ***forensisch*** stammt vom lateinischen Wort ***forum*** (Marktplatz) ab. Früher war der Marktplatz der Schauplatz von Gerichtsverfahren. Mit forensisch wird jeder Bezug zu Aspekten des Rechtssystems bezeichnet. Die forensische Wissenschaft (häufig abgekürzt als ***Forensik***) ist demnach die Anwendung wissenschaftlicher Methoden auf Fragen des Rechtssystems, etwa zur Untersuchung und Verfolgung von Straftaten.

Im üblichen Sprachgebrauch verleiht das Attribut ***wissenschaftlich*** vielen Sachverhalten eine hohe Glaubwürdigkeit. Im gleichen Zuge kann man einen Vorgang leicht diskreditieren, wenn man ihn als unwissenschaftlich bezeichnet. Viele Menschen, auch viele Wissenschaftler, verbinden Wissenschaft mit Wahrheit. In diesem Abschnitt möchten wir ein differenziertes Verständnis für Wissenschaft wecken, nämlich Wissenschaft als einen nie endenden Prozess, der nur eingeschränkt etwas mit einer universellen Wahrheit zu tun hat (unabhängig

davon, ob diese überhaupt existiert). Anschließend gehen wir auf besondere Aspekte forensischer Wissenschaften ein.

1.1.1 Die wissenschaftliche Methode

Mit Wissenschaft bezeichnen wir die Methode, mit der der Mensch versucht, die Welt um sich herum zu beschreiben und zu verstehen. Wir möchten dabei allgemeine Regeln und Prinzipien aufstellen, die die Welt erklären. Beispiele hierfür sind die Naturgesetze in der Physik, etwa der Zusammenhang zwischen Masse, Beschleunigung und Geschwindigkeit. Derartige Regeln können auf verschiedene Arten hergeleitet werden. Ein häufiger Ansatz besteht darin, dass wir wiederkehrende Muster in unserer Umwelt wahrnehmen und diese in Form allgemeiner Regeln beschreiben. Wenn man glaubt, eine allgemeine Regel entdeckt zu haben, kann man diese Regel durch Experimente prüfen. Hier schließt man vom Speziellen auf das Allgemeine. Experimente müssen dabei nicht zwangsläufig quantitative, also messbare Daten hervorbringen. Auch qualitative Daten, die durch bloße Beobachtung entstehen, können wichtig sein. Bei physikalischen Experimenten muss man beispielsweise nicht immer die Geschwindigkeit eines Objektes exakt messen, es reicht häufig aus zu beobachten, ob ein Objekt schneller ist als das andere.

Ein zentraler Bestandteil der wissenschaflichen Methode sind ***Hypothesen***. Hypothesen sind Aussagen, deren Gültigkeit untersucht werden kann. Dies geschieht beispielsweise durch Experimente. Das Aufstellen von Hypothesen ist also genauso wichtig wie ihre Überprüfbarkeit. Das zentrale Merkmal von Überprüfbarkeit ist die Falsifizierbarkeit einer Hypothese (Popper, 1962). Es muss also ein Experiment existieren, dessen möglicher Ausgang die Hypothese widerlegt. In der Realität ist es generell unmöglich, ein Prinzip als allgemeingültig nachzuweisen. Man kann höchstens daran scheitern, es zu widerlegen. Solange ein Prinzip trotz vielfacher Anstrengung nicht widerlegt wurde, wird es akzeptiert und gilt in diesem Sinne innerhalb der wissenschaftlichen Gemeinschaft als richtig. Diese Art von systematischem Zweifel garantiert eine möglichst hohe Objektivität wissenschaftlicher Erkenntnisse.

Insbesondere in forensischen Wissenschaften werden häufig Zusammenhänge als richtig dargestellt, wie etwa die Behauptung, dass die Kugel am Tatort von einer bestimmten Waffe abgefeuert wurde. Als (forensischer) Wissenschaftler muss man sich jedoch immer vor Augen halten, dass man diese Tatsache im

Sinne wissenschaftlicher Arbeit nicht beweisen kann. Nur wenn man wiederholt und mit adäquaten Methoden daran scheitert, den Zusammenhang zu widerlegen, kann man schließlich zur Überzeugung gelangen, dass die Kugel tatsächlich durch die Waffe abgefeuert wurde. In anderen Kontexten kann man durch bestimmte Tests die Wahrscheinlichkeit bestimmter Tatbestände beziffern, etwa bei der DNA-Analyse. Hier kann man auch durch den Vergleich der Wahrscheinlichkeiten bestimmter Sachverhalte einschätzen, welcher Tatbestand am wahrscheinlichsten ist. Aber selbst eine hohe Wahrscheinlichkeit bestätigt nicht die Richtigkeit einer Hypothese.

Hypothesen und Experimente müssen nachvollzogen und durch die Fachgemeinschaft begutachtet werden können. Dies ist ein weiterer wesentlicher Bestandteil der wissenschaftlichen Methode. Die Präsentation von wissenschaftlichen Ergebnissen (Hypothesen und Experimente) geschieht in der Regel auf wissenschaftlichen Konferenzen oder in Fachzeitschriften. Dabei werden die Resultate vor der Veröffentlichung durch unabhängige und anerkannte Experten bewertet. Ergebnisse, die diesen Auswahlprozess überstanden haben, werden in den Fundus des akzeptierten Wissens aufgenommen. Die Veröffentlichung dient dazu, die Ergebnisse der Fachgemeinschaft dauerhaft zugänglich zu machen. Auf dieses veröffentlichte Wissen wird in neuen Arbeiten dann in Form von Zitaten Bezug genommen. Auch haben Veröffentlichungen, die keinem Begutachtungsprozess unterliegen, wie etwa „Whitepapers“ oder Einträge in Diskussionsforen im Internet, einen deutlich geringeren wissenschaftlichen Wert. Erst durch die Aufarbeitung der Ergebnisse im Rahmen einer begutachteten Publikation werden diese wirklich von der Fachgemeinschaft wahrgenommen. Wissenschaft ist darum immer auch ein Produkt von mehreren Personen, nicht von Einzelnen. Der Stand der Technik ist das Produkt einer gemeinsamen Überzeugung, die im fachlichen Austausch entsteht. Diese reflektierte Überzeugungskraft ist ein wesentlicher Faktor, warum Wissenschaftlern hohe Glaubwürdigkeit zugemessen wird. Schließlich geht es vor Gericht auch darum, eine Person (den Richter) oder eine Menge von Personen (die ***jury***) von einem gewissen Sachverhalt zu überzeugen.

1.1.2 Zur Dauerhaftigkeit wissenschaftlicher Erkenntnisse

Die eben gemachten Ausführungen zeigen, dass der ***Stand der Wissenschaft*** immer nur die jeweils aktuell akzeptierte „Wahrheit“ wiedergibt, also die zur Zeit präziseste Beschreibung der Regeln, die die Welt regieren. Diese

Beschreibung ändert sich immer dann, wenn es neue Erkenntnisse gibt, die bestehende Prinzipien widerlegen. Die Wissenschaft ist auch von zeitlichen Strömungen und Moden abhängig, vor allem in vielen Geisteswissenschaften wie etwa der Philosophie oder der Literaturwissenschaft, wodurch ein Mehrwert gerade durch die Einbringung der eigenen Subjektivität entsteht. Dennoch ist der Wissensstand auch in diesen Bereichen nicht vollkommen beliebig. Dies liegt auch an der Natur der wissenschaftlichen Methode, die eher dazu tendiert, Erkenntnisse zu ***verfeinern*** statt sie grundlegend zu widerlegen. Einige historische Ausnahmen („Die Erde ist eine Scheibe.“) bestätigen die Regel.

Wie weiter unten deutlich werden wird, gibt es auch in den forensischen Wissenschaften zwar immer wieder große Umbrüche, die unser Verständnis für bestehende Sachverhalte grundlegend verändern. Die Entdeckung verschiedener Blutgruppen etwa erlaubte es, die Zuordnung von Blutspuren zu bestimmten Personen auszuschließen. Zuletzt revolutionierte die Akzeptanz von DNA-Tests vor Gericht die Arbeit von Ermittlern. DNA-Tests sind aber auch gleichzeitig ein Beispiel für die erstaunliche Dauerhaftigkeit des wissenschaftlichen Fortschritts, denn erst mit der Einführung dieser Tests konnten Blutspuren von Personen mit der gleichen Blutgruppe unterschieden werden. Erkenntnisse, die allein auf Blutgruppen basierten, wurden nicht plötzlich falsch, denn unterschiedliche Blutgruppen ergeben auch unterschiedliche DNA-Profile. Durch den neuen Test konnten jedoch andere und genauere Unterscheidungen getroffen werden als vorher.

1.1.3 Forensische Wissenschaften, Kriminalistik und Kriminologie

Man unterscheidet häufig zwischen reinen und angewandten Wissenschaften. Reine Wissenschaften versuchen, Erkenntnisse über die Welt um ihrer selbst Willen zu erzielen. Ein gutes Beispiel ist die reine Mathematik, die die Gesetzmäßigkeiten von Zahlen untersucht, ohne eine konkrete Anwendung im Blick zu haben. Forensische Wissenschaften sind jedoch in der Regel angewandte Wissenschaften – ähnlich wie weite Bereiche der Medizin oder der Ingenieurswissenschaften. Der Grund hierfür wurde eingangs erwähnt und wird später noch eingehed erläutert: Ausgangspunkt der Arbeit eines forensischen Wissenschaftlers ist in der Regel immer eine juristische Fragestellung, deren Beantwortung durch die wissenschaftlichen Methoden einer Disziplin unterstützt

werden soll. Dies manifestiert sich auch darin, dass forensische Wissenschaftler eher analysieren als experimentieren.

Dennoch ergeben sich immer wieder auch Fragestellungen, die die Natur forensischer Wissenschaften an sich oder einer speziellen forensischen Wissenschaft betreffen. Und grundsätzliche Untersuchungen in einem Fach, etwa zu mechanischen Ursache/Wirkungsbeziehungen in der Physik oder zur prinzipiellen Lösbarkeit algorithmischer Probleme in der Informatik, helfen auch immer, den Einsatz der jeweiligen wissenschaftlichen Methoden für forensische Zwecke zu verbessern. Die Inhalte dieses Kapitels etwa, das versucht, das gesamte Gebiet der forensischen Wissenschaften zu hinterfragen und zu systematisieren, ist eher Ausdruck eines allgemeinen Erkenntnisinteresses als angewandter Forschung.

Im späten 19. Jahrhundert prägte Hans Groß den Begriff ***Kriminalistik*** für die damals neu aufkommenden Untersuchungsmethoden der Polizei. In seinem ***Handbuch der Kriminalistik*** (Groß u. Geerds, 1977) entwickelte Groß den Begriff zunächst als Oberbegriff für alle Arten von Polizeiwissenschaften, nämlich als „Lehre von der [. . .] Bekämpfung der Kriminalität durch die Strafverfolgungsorgane [. . .] in der Lebenswirklichkeit“ (Groß u. Geerds, 1977, S. 5). Auch Groß grenzte seinen Begriff von den rechtswissenschaftlichen Fragestellungen ab, die mit der Bekämpfung von Kriminalität einhergehen (Fragen des Strafrechts und des Strafprozessrechts). Im Fokus des Interesses standen für ihn aber immer die einzelne Straftat sowie die technischen und organisatorischen Maßnahmen, die man ergreifen muss, um diese Tat aufzuklären oder zu verhindern. Hierunter fallen vier Bereiche (Groß u. Geerds, 1977, S. 8):

1. Die ***Verbrechenstechnik***, also die Phänomenologie kriminellen Verhaltens, legt ein starkes Gewicht auf die ***mehr handwerkliche Seite der Tatausführung*** und weniger etwa auf die Beweggründe des Täters. Man spricht in diesem Zusammenhang auch vom ***Modus Operandi***.
2. Die ***Kriminaltechnik*** zielt auf die Erbringung von ***Sachbeweisen*** zur Aufklärung von Straftaten. Traditionell geht es hierbei um die Sicherung und Analyse physischer Spuren wie etwa Blutspuren, Schmauchspuren oder Fingerabdrücke.
3. Während die Kriminaltechnik die geeigneten Mittel und Untersuchungstechniken bereitstellt, geht es bei der ***Kriminaltaktik*** darum, wie man diese Mittel in einem größeren Zusammenhang sinnvoll und

ökonomisch anwendet. In diesen Bereich gehören beispielsweise die Tatortarbeit, die Fahndung und die Vernehmung. Dabei muss auch die Frage nach der Zulässigkeit bestimmter Untersuchungsmethoden in einem gegebenen Kontext betrachtet werden.

4. Schließlich gibt es noch die ***Organisation der Verbrechensbekämpfung*** insgesamt, also die Art und Weise, wie etwa Aufgaben zwischen Behörden und einzelnen Personen aufgeteilt sind und wie sie zusammenarbeiten. Dazu gehören die Zusammenarbeit zwischen Polizei, Staatsanwaltschaft und Gerichten sowie die internationale Kooperation zwischen den Strafverfolgungsbehörden.

Die Kriminalistik orientiert sich insgesamt eher an den Natur- und Ingenieurswissenschaften und grenzt sich dadurch mehr oder weniger scharf von der ***Kriminologie*** ab, also der Lehre von den Ursachen und den Erscheinungsformen der Kriminalität, die eher den Sozialwissenschaften und der Psychologie zugeordnet wird.

Der Begriff der Kriminalistik wird heute mit unterschiedlichen Bedeutungsnuancen verwendet. Hervorzuheben ist jedoch, dass immer eine einzelne Straftat und die eher handwerkliche Seite der Tatausführung im Zentrum des Interesses steht und dass sowohl repressive als auch präventive Maßnahmen darunter fallen (Weihmann, 2007).

Exkurs 1 (Die „sozialistische“ Kriminalistik) ***Aus deutscher Sicht ist bemerkenswert, dass sich der Kriminalistikbegriff nach dem zweiten Weltkrieg in den beiden deutschen Staaten unabhängig und zum Teil sehr unterschiedlich entwickelte. So wurden zentrale Begriffe wie*** Spur, Individualisierung, Assoziation ***und*** Transfer (***auf die später noch detailliert eingegangen wird) in der „sozialistischen Kriminalistik“ (Schurich u. Scharf, 1979) der DDR unabhängig von der westlichen Literatur entwickelt (siehe etwa Schurich (1974, 1982); Schurich u. Scharf (1979)). Bis in die 1990er Jahre war die Kriminalistik zudem an der Humboldt Universität zu Berlin als eigenständiges wissenschaftliches Fach mit mehreren Professuren vertreten, während in der Bundesrepublik die Kriminalistik allenfalls als Teilgebiet der Rechtswissenschaften in Erscheinung trat. Problematisch erscheint hingegen der Begriff der*** Version, ***der heute oft mit dem Begriff der Hypothese gleichgestellt wird, in der DDR jedoch einen Wahrheitsanspruch hatte, der dem oben skizzierten (und hier angenommenen) Wissenschaftsbild widerspricht (Weihmann, 2008***).

Wir werden den Begriff der Kriminalistik im Folgenden meiden und stattdessen von forensischen Wissenschaften sprechen. Hierdurch treten einerseits die Bezüge zu anderen Wissenschaften wie der Biologie oder der Informatik deutlicher zu Tage. Andererseits reflektiert dies auch den angelsächsischen Sprachgebrauch vor allem in den USA, wo eher von ***forensic science*** gesprochen wird und selten von ***criminalistics***. Allerdings tritt dadurch die Beziehung zwischen ermittelnder Polizeiarbeit und Spurenanalyse etwas in den Hintergrund, was gerade von Vertretern einer „allgemeinen“ forensischen Wissenschaft kritisch gesehen wird (Margot, 2011). Wie oben bereits erwähnt, tut man gut daran, die eigenen wissenschaftlichen Erkenntnisse nicht als einzig wahr zu beurteilen; was (eine) Wissenschaft ist und was nicht, hängt auch von gesellschaftlichen Entwicklungen ab und kann sich ändern. Wesentlich ist das Erkenntnisinteresse der beteiligten Personen, ein Phänomen besser zu verstehen oder eine Straftat aufzuklären.

1.1.4 Forensische Wissenschaft und das Rechtssystem

Ohne das Rechtssystem gäbe es keine forensischen Wissenschaften. Die Ziele und Fragestellungen beider Bereiche sind jedoch grundsätzlich verschieden. In einem Rechtsstreit geht es ausschließlich darum, rechtliche Fragen zu beantworten. Diese beziehen sich darauf, ob ein Rechtsverstoß stattgefunden hat und falls ja, welcher. Schlussendlich kann die rechtlichen Fragen nur das Rechtssystem bzw. der Richter beantworten. Die Aufgabe forensischer Gutachter ist es, die rechtlichen Fragen in wissenschaftliche Fragen zu übersetzen. Wenn das nicht möglich ist, sind forensische Wissenschaftler fehl am Platze. Die Frage, ob ein Beschuldigter tatsächlich einen Mord begangen hat, kann beispielsweise in die Frage übersetzt werden, welche genetischen Sequenzen in den Blutspuren am Tatort gefunden werden können. Eine Antwort auf die wissenschaftliche Frage kann anschließend helfen, eine Antwort auf die rechtliche Frage zu finden. Allerdings hilft nicht jede wissenschaftliche Frage bei der Wahrheitsfindung. Der Nachweis, dass sich Blutspuren des Tatverdächtigen an seinen eigenen Schuhen befinden, hilft nicht notwendigerweise bei der Beantwortung der Frage, ob der Tatverdächtige den Mord begangen hat. Genauso wenig hilft es herauszufinden, dass sich auf der Tatwaffe mikroskopische Fasern aus der Kleidung des Opfers befinden, wenn der Tatverdächtige Kleidung trägt, die auch aus diesen Fasern besteht.

Übersetzt man die rechtliche Frage in eine wissenschaftliche Frage, verliert man jeden Bezug zu den Begriffen Schuld und Unschuld. Forensische Wissenschaften suchen immer nach Verbindungen zwischen Objekten, beispielsweise zwischen dem Blut, das sich am Tatort und der Tatwaffe befindet, und dem Blut des Opfers und des Tatverdächtigen. Über Schuld oder Unschuld entscheidet ausschließlich das Gericht.

1.1.5 Frühe literarische Bezüge

Seit ihren Anfängen umgibt die forensischen Wissenschaften eine Aura von Geheimnis und Zauberei. Dazu beigetragen haben auch die forensischen Praktiker, welche die Fähigkeit, von wenigen Spuren am Tatort nahezu den kompletten Fall zu rekonstruieren, als Ausdruck von Scharfsinnigkeit und Talent etablierten. Leider führte dies dazu, dass es immer schwieriger wurde, echte Experten von Scharlatanen zu unterscheiden, die mitunter andere Interessen als die Wahrheitsfindung hatten (Inman u. Rudin, 2000, S. 22). Die Selbstüberschätzung von forensischen Experten basiert vermutlich teilweise auf den Anfängen der Kriminalliteratur, die viele Fortschritte bei den polizeilichen Ermittlungsmethoden begeistert aufnahm und dramaturgisch überhöhte.

Das bekannteste Beispiel ist der geniale Spurenleser und Detektiv Sherlock Holmes, der im London des späten 19. und frühen 20. Jahrhunderts Straftäter mit Scharfsinnigkeit und einem Vergrößerungsglas überführt. Beispielsweise identifiziert Holmes in ***The Red-Headed League*** (deutscher Titel: Der Bund der Rotschöpfe) die gerauchte Zigarettenmarke anhand von Aschespuren oder die Herkunft einer Person anhand von Dreckspuren am Mantel. Sir Arthur Conan Doyle (1859–1930), der literarische Vater dieser Detektivgeschichten, war bekannt dafür, dass er aktuelle Fortschritte der Polizeiwissenschaften regelmäßig studierte und in seine Romane integrierte. In ***A Study in Scarlet*** (deutscher Titel: Eine Studie in Scharlachrot) erfindet Holmes etwa einen verbesserten Test zum Nachweis von Blut. Auch forensische Wissenschaftler, wie der Franzose Edmund Locard, nehmen in ihren Arbeiten direkten Bezug auf Fälle aus der Holmes-Literatur (Inman u. Rudin, 2000, S. 24).

In der Literaturwissenschaft gilt nicht Doyle, sondern Edgar Allen Poe (1809–1849) als der Erfinder des Genres der Detektivromane. In ***A Study in Scarlet*** bezieht sich Doyle direkt auf die durch Poe eingeführte Detektivfigur des Auguste Dupin. Typisch für die allwissende Attitüde der Holmes'sehen Figur

disqualifiziert dieser seinen literarischen Vorgänger als ***minderwertig*** (***inferior***). Gerade diese Haltung, welche die Person Holmes so interessant macht, ist leider ein schlechtes Vorbild für moderne forensische Experten, denen Bescheidenheit besser ansteht als Angeberei.

Auch Mark Twain, der nicht gerade für Detektivgeschichten bekannt ist, verfolgte die Entwicklungen der Polizeiwissenschaften mit großem Interesse. In ***The Tragedy of Pudd'nhead Wilson***, geschrieben 1894, erzählt er die Geschichte eines Anwalts, der zwei Brüder vor der Verurteilung rettet, indem er nachweist, dass die blutigen Fingerabdrücke auf einem Messer nicht von diesen stammen (Inman u. Rudin, 2000, S. 28). Mit der detaillierten Beschreibung der Abdrücke und deren charakteristischen Eigenschaften ist Twain seiner Zeit voraus, denn es sollten noch einige Jahre ins Land gehen, bevor Fingerabdrücke als identifizierendes Merkmal vor Gericht akzeptiert wurden.

1.1.6 Analyse von Spuren zur Identifizierung von Personen

Die Besonderheit von Fingerabdrücken ist spätestens seit dem 17. Jahrhundert bekannt, aber ihr konkreter Nutzen bei der Identifizierung von Personen ist erst im späten 19. Jahrhundert deutlich geworden. Bis dahin galt die ***Anthropometrie*** als einzige Möglichkeit, die Identität einer Person eindeutig zu überprüfen. Dabei mussten mehrere Bereiche des Körpers genau vermessen werden. Die Kombination dieser Messwerte sei ein individualisierendes Merkmal, behauptete jedenfalls Alphonse Bertillon, der 1883 mit mehreren spektakulären Identifizierungen erst in Frankreich und dann weltweit dieser Methode zum Durchbruch verhalf (Inman u. Rudin, 2000, S. 30).

Als im Jahre 1903 in einem Gefängnis in Kansas zwei Personen mit identischen anthropometrischen Merkmalen nur aufgrund ihrer unterschiedlichen Fingerabdrücke identifiziert werden konnten, war das Ende der Anthropometrie gekommen, und Fingerabdrücke wurden das primäre identifizierende Merkmal. Wegbereiter für die Wissenschaft der ***Daktyloskopie*** waren vor allem der englische Naturwissenschaftler Francis Galton der 1892 ein Buch über die Aussagekraft und die statistischen Merkmale von Fingerabdrücken veröffentlichte, sowie der argentinische Kriminologe Ivan Vučetić (auch: Juan Vucetich), der ein standardisiertes Klassifikationssystem für Fingerabdrücke erfand. Seitdem sind vor allem bei den Möglichkeiten der Sicherung von Fingerabdrücken Fortschritte erzielt worden.

Ein weiterer Meilenstein bei der Analyse von Spuren war die Entdeckung verschiedener Blutgruppen um 1900 durch Karl Landsteiner, die erstmals die grobe Zuordnung von Blutspuren zu Personen erlaubte. Wenn eine Blutspur eine andere Blutgruppe hatte als eine bestimmte Person, konnte diese Person als Quelle dieser Spur ausgeschlossen werden. Durch einen von Pauf Uhlenhuth 1901 entwickelten Test war es zudem möglich Tier- und Menschenblut eindeutig zu unterscheiden. Damit konnte die damals übliche Schutzbehauptung entkräftet werden, dass es sich bei bestimmten Spuren um Tierblut handle.

Ähnliche Fortschritte gab es im Verlauf des 20. Jahrhunderts auf vielen Gebieten der Medizin. Als Meilenstein für die Identifizierung von Personen gilt jedoch die DNA-Analyse, die in den 1980er Jahren von Sir Alex Jeffreys entscheidend vorangetrieben wurde. Der ***DNA-Fingerabdruck*** wird heute mit der gleichen Überzeugung als individualisierendes und identifizierendes Merkmal aufgefasst wie fast 100 Jahre zuvor der Fingerabdruck.

1.2 Spuren und ihre Entstehung

Eine Spur wird als ein ***hinterlassenes Zeichen*** angesehen. Das Wort ***Spur*** kommt ursprünglich aus dem Altgermanischen und bedeutet dort ***Tritt*** oder ***Fußabdruck***. In der Jägersprache spricht man häufig auch von ***spuren*** und meint dabei das Ansetzen eines Hundes auf die Fährte (Weihmann, 2007).

Wir verwenden hier einen sehr allgemeinen Spurenbegiff und verbleiben zunächst in der physischen Welt, betrachten also physische (im Sinne von nicht-digitalen) Spuren. Eine ***Spur*** (***evidence***) ist dabei jedes Objekt, was ein Argument plausibler macht. In der juristischen Fachsprache spricht man statt von Spur auch häufig von ***Beweismittel***. Eine Spur ist demnach ein Gegenstand zusammen mit einer Reihe von dokumentierten Behauptungen (***claim***) oder Annahmen (***assumption***) über diesen Gegenstand. Diese Behauptungen beschreiben im Wesentlichen die ***Herkunft*** (***provenance***) der Spur, etwa den Fundort und die Zeit des Auffindens.

In der realen Welt kann nahezu alles zur Spur werden. Weihmann (2007) verzeichnet in seinem Buch mehr als ein Dutzend verschiedener Arten von Spuren. Im Handbuch von Groß u. Geerds (1977) umfasst der Abschnitt zu Spurenkunde und deren Untersuchung mehr als 200 Seiten. Die Palette reicht von Blut, Haaren und Sekreten über Wasser, Boden, Vegetation, Luft und Gas bis hin zu Metall, Kunststoff, Holz und Textilien. Etwas wird zur Spur, wenn es eine Beziehung zum untersuchten Sachverhalt, z.B. der Straftat, aufweist (Kirk,

1974, S. 6). Kirk (1974, S. 1 ff.) macht deutlich, wie wichtig derartige Spuren bei Ermittlungen sein können, denn es ist nahezu unmöglich, Handlungen auszuführen, ohne irgendwelche Spuren zu hinterlassen. Kirk (1974, S. 2) schreibt:

> „Not only his fingerprints and his shoeprints, but also his hair, the fibers from his clothes, the glass he breaks, the tool marks he leaves, the paint he scratches, the blood or semen he deposits or collects – all these bear mute witness against him."

Wir betrachten im Folgenden allgemeine Eigenschaften von Spuren und die Prinzipien, die bei der Entstehung solcher Spuren eine Rolle spielen.

1.2.1 Spureninformation und Spurenträger

Betrachten wir zur Illustration als stereotype Spur einen blutigen Handschuh, der am Tatort eines Tötungsdelikts gefunden wird. Wenn an dem Handschuh DNA-Spuren des Täters zu finden sind, dann wird der Handschuh zur Spur. Hilfreich ist der blutige Handschuh, weil mit ihm die Verbindung von Täter zu Tatort gezogen werden kann. Diese Verbindung entsteht jedoch genau genommen aus den Informationen, die innerhalb der DNA gespeichert sind. Die DNA selbst (und schlussendlich auch der blutige Handschuh) sind nur die Träger dieser Information.

Allgemein unterscheidet man bei Spuren zwischen ***(Spuren-)Information (information***) und ***(Spuren-)Träger*** (***support***). Die Information ist die „Bedeutung" der Spur, also das, was sie aussagt. Der Träger ist diejenige Materie, die diese Bedeutung trägt. Ein anderes Beispiel für den Unterschied zwischen Information und Träger ist ein Brief, der am Tatort gefunden wird. Der Träger ist das Papier und die Tinte, aus denen der Brief besteht. Die Information ist (unter anderem) die auf dem Brief codierte Bedeutung der Schriftzeichen.

In der klassischen Forensik ist die Unterscheidung zwischen Spurenträger und Spureninformation recht unintuitiv, weil eine starke Verbindung besteht zwischen Träger und Information (wie im Beispiel DNA zu Blut oder Text zu Brief): Information und Träger werden dort leicht als Einheit betrachtet, die nicht aufgetrennt werden kann. Aber schon bei einem Brief ist diese Verbindung schon etwas weniger stark. Betrachtet man beispielsweise ein Erpressungsschreiben, dann ist im Wesentlichen die Bedeutung der

geschriebenen Worte (also die Spureninformation) für den Fall relevant und es ist egal, mit welcher Schreibmaschine, mit welcher Tinte oder auf welchem Briefpapier der Brief geschrieben wurde. Der Spurenträger kann jedoch weitere Informationen enthalten, etwa die Form der (Hand-) Schrift, die benutzt werden kann, um die Spur einer Person oder einer Schreibmaschinen zuzuordnen. Wir werden diesen Unterschied in Kapitel 2, in dem es um digitale Spuren geht, noch klarer sehen.

Alle physischen Spuren sind — wie jede Materie — vergänglich. Genauer gesagt sind die Spurenträger vergänglich, auch wenn der Grad der Vergänglichkeit sehr unterschiedlich sein kann. Bei physischen Spuren haben wir heute ein recht gutes Verständnis von ihrer Vergänglichkeit.

1.2.2 Integrität und Authentizität von Spuren

Ermittler suchen am Tatort nach Spuren, die über den Tathergang Aufschluss geben können. Insbesondere Spuren, die unabsichtlich hinterlassen werden, wie etwa Fingerabdrücke, Fasern oder Sekrete, sind für die Ermittler interessant. Diese sind jedoch in der Regel schwer zu erkennen und müssen von Spezialisten gesucht und aufgenommen werden. Dieser Vorgang wird allgemein auch ***Spurensicherung*** genannt. Bei der Sicherung einer Spur entstehen die ersten Behauptungen, die in irgend einer Form an die Spur geheftet werden, etwa als Aufschrift auf einem Asservatenbeutel (***evidence bag***) oder als Eintrag in einem Durchsuchungsbericht.

Spuren können eine Theorie über einen Tathergang stützen oder widerlegen. Die Überzeugungskraft solcher Spuren hängt stark davon ab, wie „nahe“ die (vor Gericht) vorgelegten Spuren an den Spuren sind, die am Tatort gefunden wurden. Allgemein versucht man diese Nähe mit zwei Begriffen zu fassen, nämlich mit der ***Integrität*** und ***Authentizität*** von Spuren:

- ***Integrität*** bedeutet grob gesprochen, dass die Spur seit ihrer Sicherung nicht verändert wurde. Integrität ist aber keine Eigenschaft, die sofort verloren geht, wenn sich die Spur verändert. Da alle Materie vergänglich ist, wird es auch nach kurzer Zeit schon Veränderungen am gesicherten blutigen Handschuh geben (beispielsweise trocknet das Blut). Veränderungen können sogar durch den Sicherungsprozess geschehen. Die Veränderungen sollten jedoch so gering wie möglich gehalten werden. Integrität bezieht sich hauptsächlich auf die ***Spureninformation*** und stellt

somit sicher, dass die Information unverändert vorliegt.

- ***Authentizität*** bedeutet, dass die vorgelegte Spur tatsächlich die Spur ist, die durch die mit ihr verbundenen Behauptungen beschrieben wird. Wenn von der Spur behauptet wird, dass es sich dabei um den blutigen Handschuh vom Tatort handelt, dann ist die Spur authentisch, wenn es sich tatsächlich um ***den*** blutigen Handschuh vom Tatort handelt. Bei physischen Objekten definiert sich Authentizität also hauptsächlich durch den ***Spurenträger***.

Beispiel 1 (Authentizität vs. Integrität) ***Um den Unterschied zwischen Authentizität und Integrität darzulegen, nehmen wir das oben eingeführte Beispiel des blutigen Handschuhs vom Tatort und betrachten zwei Fälle:***

1. ***Angenommen die Spur behauptet, sie ist der blutige Handschuh vom Tatort, der*** unter ***der Leiche gefunden wurde. In Wirklichkeit ist sie ist aber der blutige Handschuh vom Tatort, der*** auf ***der Leiche gefunden wurde. Die Spur erfüllt (falls sie nicht verändert wurde) zwar Integrität aber nicht Authentizität***.

2. ***Falls die Spur behauptet, der blutige Handschuh zu sein, der auf der Leiche gefunden wurde, und es sich tatsächlich um diesen Handschuh handelt, der jedoch nach der Sicherung einmal mit der Waschmaschine gewaschen worden ist, dann erfüllt die Spur Authentizität aber nicht Integrität (jedenfalls nicht in hohem Maße***).

Beide Begriffe machen also nur Sinn, wenn eine Spur als Einheit von Information und Träger verstanden wird.

Nur die Kombination von Integrität und Authentizität erlaubt es sicherzugehen, dass die Spur tatsächlich sinnvoll und hilfreich für die Wahrheitsfindung interpretiert werden kann. Spuren müssen also authentisch sein und ein Mindestmaß an Integrität besitzen. Wenn die Integrität verloren geht (bzw. die Integrität unter ein bestimmtes Maß sinkt), dann kann man bestimmte Schlüsse nicht mehr aus den durch die Spur dargelegten Informationen ziehen: Wenn der Handschuh gewaschen worden ist, ist möglicherweise die DNA des Täters verschwunden (ganz zu schweigen von der Möglichkeit, dass fremde DNA auf den Handschuh gelangt ist). Ähnliches gilt für die Authentizität. Die Sicherung und spätere Behandlung von Spuren muss

also so erfolgen, dass die Spuren möglichst wenig verändert werden. Um dies zu gewährleisten, wird der Tatort in der Regel großräumig abgesperrt und darf nur von befugten Personen betreten werden. Diese Personen tragen zudem häufig Schutzanzüge, um nicht versehentlich selbst Spuren am Tatort zu hinterlassen.

Authentizität und Integrität beruhen stark auf Annahmen, etwa dass die mit der Spur betrauten Personen ordnungsgemäß und verlässlich gearbeitet haben. Neben der Gefahr einer Modifikation bei der Spurensicherung müssen im weiteren Verlauf der Untersuchung alle weiteren Gefahren, welche die Spur verändern können, eingedämmt werden. Dies kann beispielsweise dadurch erreicht werden, dass möglichst wenige und auch nur fachlich besonders qualifizierte Personen Zugang zu der Spur haben. Dies wird in der so genannten ***Verwahrungskette*** (***chain of custody***) dokumentiert. Die Verwahrungskette ist Teil der Dokumentation der Spurenherkuft, also Teil der mit der Spur verbundenen Behauptungen. Sie dokumentiert lückenlos, wann welche Personen Zugang zur Spur hatten. Wenn keine Personen Zugang zur Spur haben sollen, wird diese in einer besonders geschützten Umgebung (Asservatenkammer) aufbewahrt. Aber auch hier muss man annehmen (und darauf vertrauen), dass die Verwahrung ordnungsgemäß und verlässlich organisiert worden ist.

Nach der Sicherung werden die Spuren näher untersucht. Diesen Vorgang nennt man ***Spurenanalyse***, und er geschieht meist durch Spezialisten in speziellen Laboren. In der Praxis spricht man auch von ***kriminaltechnischer Untersuchung***. Dort werden die Spuren durch Experten, in der Regel Wissenschaftler (Biologen, Physiker, Chemiker, Mediziner oder Informatiker) analysiert.

1.2.3 Klassifikation von Spuren

Nachdem fast jede Art von Materie zur Spur werden kann, gibt es auch viele verschiedene Möglichkeiten, physische Spuren zu klassifizieren. Lee u. Harris (2000, S. 5ff) zählen allein sieben verschiedene Klassifikationsarten auf. Am häufigsten findet man die Klassifikation nach Straftat (Lee u. Harris, 2000; Groß u. Geerds, 1977), aber auch stärker naturwissenschaftlich orientierte Klassifikationsschemata werden verwendet, etwa die Klassifikation nach der chemischen oder biologischen Zusammensetzung.

Lee u. Harris (2000, S. 5) unterscheiden darüber hinaus noch ***transiente*** Spuren: Das sind physische Spuren, die nur temporär existieren und demnach

leicht verloren gehen können. Beispiele für transiente Spuren sind Gerüche, etwa von Benzin, Parfüm, Rauch oder Urin. Weitere Beispiele sind Temperaturen von Gegenständen, etwa die Temperatur eines Motors, die Wassertemperatur in einer Badewanne oder die Körpertemperatur einer Leiche. Transiente Spuren müssen ohne Zeitverzug entdeckt und wenn möglich auch gesichert werden. Da die Sicherung der Spur aufgrund ihrer Natur häufig schwer bis unmöglich ist, kann man die Existenz und die Beschaffenheit der Spuren nur schriftlich festhalten und von anderen Beobachtern am Tatort bestätigen lassen. Manche transiente Spuren können auch fotografiert oder auf Video aufgezeichnet werden.

Eine Spur weist notwendigerweise eine Beziehung zum untersuchten Sachverhalt auf. Bei vielen Gegenständen, wie beispielsweise bei einer Schusswaffe oder einem Brecheisen, ist diese Beziehung deutlich erkennbar. Kirk (1974, S. 6) weist jedoch darauf hin, dass die nützlichsten physischen Spuren so klein sind, dass man sie mit bloßem Auge nicht erkennt. Kirk nennt diese Klasse von Spuren ***mikroskopische Spuren***. Die Besonderheit von mikroskopischen Spuren ist, dass sie sehr leicht übersehen werden (durch den Täter wie den Ermittler) und sehr schwer zu entfernen sind. Oft werden noch Monate oder Jahre nach einer Tatausführung mikroskopische Spuren am Tatort gefunden. Die Verfügbarkeit mikroskopischer Spuren ist also in der Regel höher als die Verfügbarkeit von großen, sichtbaren Spuren. So mag es möglich sein, die Beziehung zwischen einem Fahrzeug und dem Tatort anhand von sichtbaren Reifenspuren nachzuweisen. Viel wahrscheinlicher (und wirksamer) ist es aber, dass man eine Beziehung zwischen einem Opfer und dem Fahrzeug aufgrund von Haaren, Fasern oder Sekreten herstellen kann. So gehört ein Staubsauger mit einem speziellen Filter (siehe Abbildung 1.1) zum Standardwerkzeug der Spurensicherung am Tatort. Die aufgenommenen Partikel werden in einer Plastiktüte aufbewahrt und im Labor unter dem Mikroskop oder mit sonstigen Methoden analysiert.

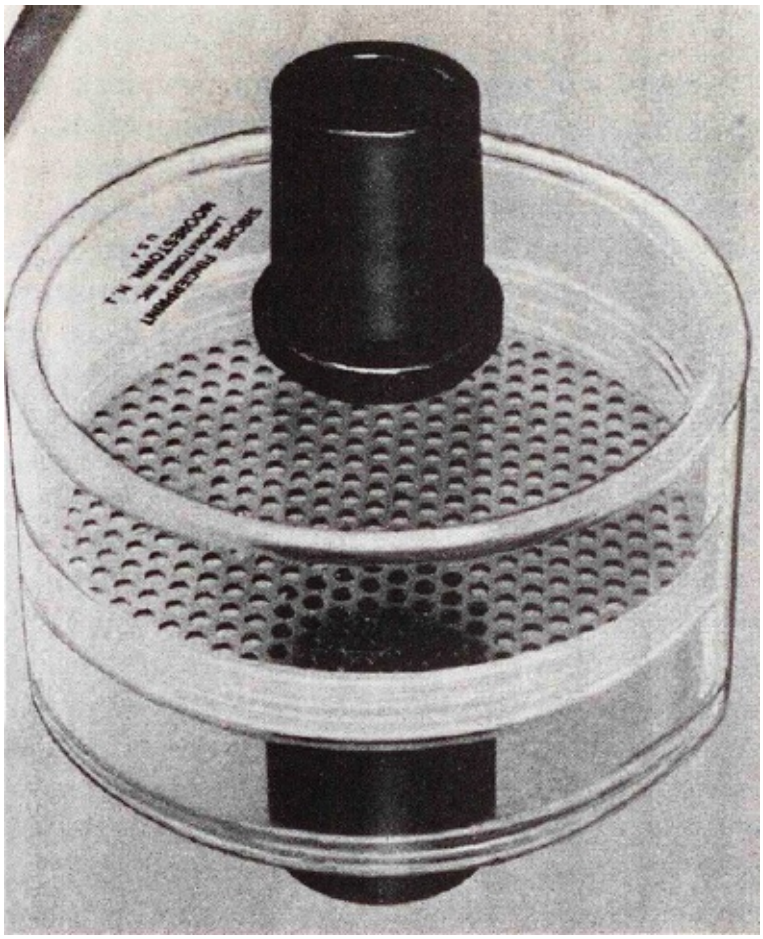


Abbildung 1.1: Filter zur Sammlung mikroskopischer Spuren (Kirk, 1974, S. 25).

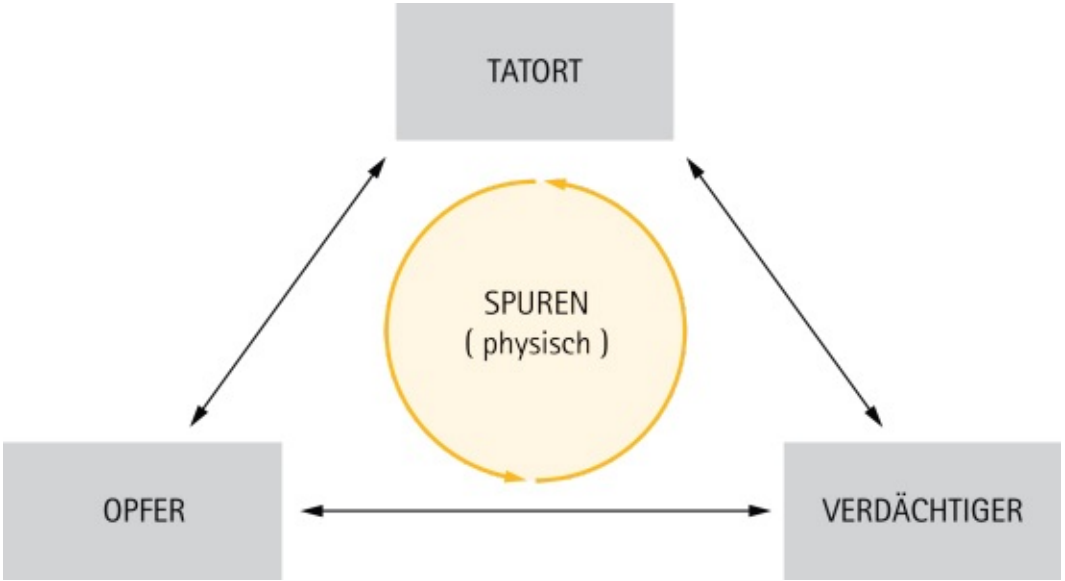


Abbildung 1.2: Schematische Darstellung von Locards Austauschprinzip.

1.2.4 Übertragung (Transfer)

Edmund Locard beschrieb als Erster eines der Grundprinzipien bei der Entstehung von Spuren. Inspiriert durch die Arbeiten von Groß und den Abenteuern von Sherlock Holmes begann er Anfang des 20. Jahrhunderts detaillierte Untersuchungsreihen zu mikroskopischen Spuren wie Staub oder Schmutz. In einem seiner Werke (Locard, 1920) (zitiert nach Inman u. Rudin (2000, S. 44)), die auf Französisch erschienen, dokumentiert er die Beobachtung, dass niemand eine Straftat begehen kann, ohne zahlreiche Zeichen (***marques multiples***) zu hinterlassen, entweder dadurch, dass er etwas am Tatort hinterlässt oder dadurch, dass er etwas vom Tatort mitnimmt. Aus dieser Beobachtung entstand das auch populärwissenschaftlich gut dokumentierte und nach ihm benannte ***Locardsche Austauschprinzip*** (***Locard's exchange principle***). Es wird häufig als Dreieck zwischen Täter, Opfer und Tatort visualisiert (siehe Abbildung 1.2).

Heute basiert ein wesentlicher Teil der Theorie forensischer Wissenschaften auf den Einsichten von Locard. Das Austauschprinzip wurde jedoch bis heute zu einer allgemeineren Theorie der ***Übertragung*** weiterentwickelt. Heute sieht man die Übertragung von Materie als einen wesentlichen, aber nicht als den einzigen Weg an, wie Spuren entstehen. Auf die Grundlagen dieser Theorie gehen wir im Folgenden ein.

1.2.5 Zerteilbarkeit (Divisibility)

Bei genauerer Betrachtung basiert das Locardsche Austauschprinzip auf einem weiteren Prinzip: Austausch passiert nämlich nur dann, wenn sich Objekte in kleinere Teile zerteilen lassen (Inman u. Rudin, 2000, Kapitel 4). Da alle Objekte in irgendeiner Form aus zerteilbarer Materie bestehen, erscheint diese Voraussetzung häufig als selbstverständlich und trivial. Wenn Objekte aber nicht zerteilbar wären, gäbe es auch keinen Austausch.

Die Materie, aus der Objekte bestehen, wird nach Regeln der Biologie, Chemie und Physik zusammengehalten. Wenn Kräfte auf ein Objekt einwirken, dann kann es dazu kommen, dass sich das Objekt in seine Einzelteile zerteilt. Die Einzelteile behalten aber in der Regel charakteristische Eigenschaften des ursprünglichen Objekts. Diese Eigenschaften können chemischer oder biologischer Natur sein. Abbildung 1.3 zeigt aber auch ein Beispiel dafür, dass auch charakteristische Muster erhalten bleiben können, wie etwa Texturen.

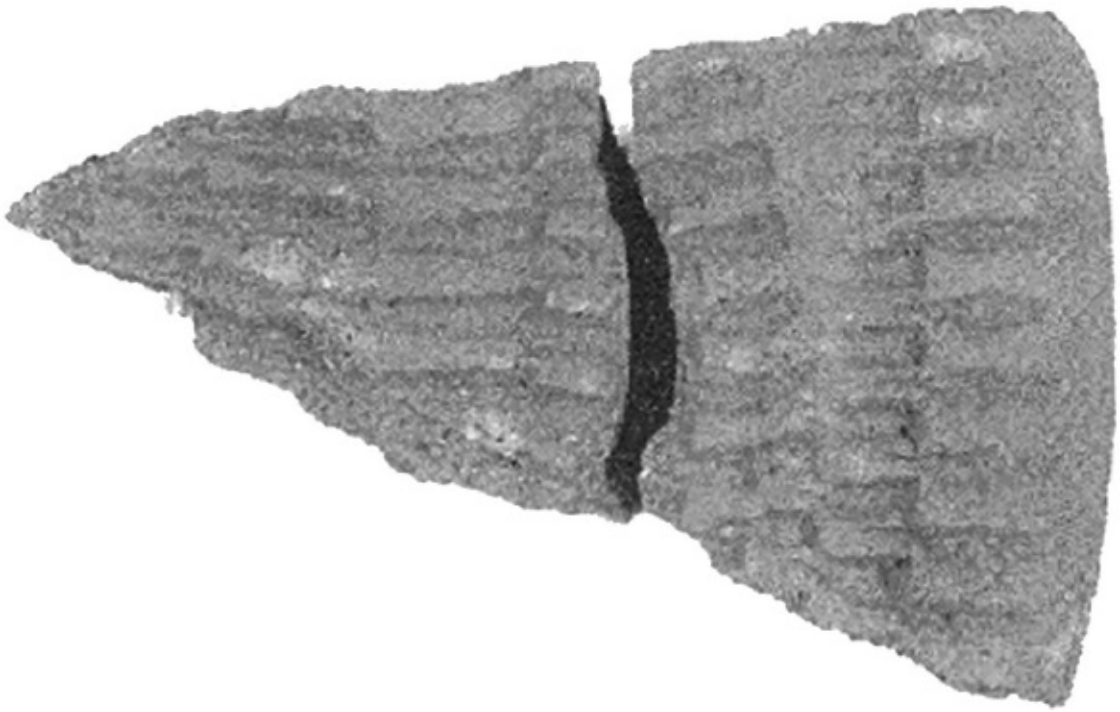


Abbildung 1.3: Transfer von Texturen auf Bruchstücke eines Objekts (Inman u. Rudin, 2000, S. 89).

Manchmal entstehen durch das Zerteilen auch neue charakteristische Eigenschaften. Als Beispiel werden häufig Riss- oder Bruchkanten angeführt (siehe Abbildung 1.4).

1.2.6 Übertragung von Mustern

„Austausch“ im Sinne des Locardschen Austauschprinzips war immer eine Übertragung von Materie. Durch Abdrücke entsteht jedoch auch eine Form von Austausch, die nicht notwendigerweise Materie überträgt. Beispiele für diese Art von Austausch sind Fußabdrücke oder Reifenspuren. Auch wenn hierbei häufig auch Materie übertragen wird (beispielsweise Erde am Schuh oder am Reifen), so ist diese Materie nach längerer Zeit kaum mehr nachweisbar. Es bleiben die Muster, die sowohl am Schuh bzw. am Reifen als auch im Abdruck existieren.

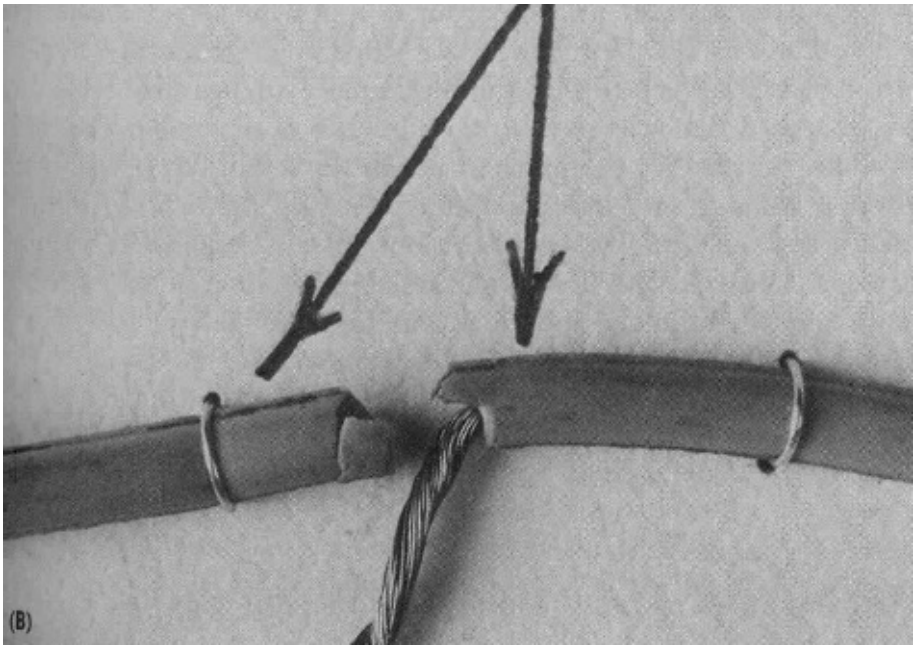


Abbildung 1.4: Übereinstimmende Bruchkanten in einem Kabel (Kirk, 1974, S. 139).

Die Übertragung von Mustern wird auch mit ***transfer of traits*** bezeichnet (Inman u. Rudin, 2000, S. 93). Das Zerteilen von Objekten ist also nicht bei jedem Austausch notwendig: Wenn Materie übertragen wird, ist Zerteilung eine notwendige Voraussetzung; wenn Muster übertragen werden, dann nicht. Wir betrachten nun zwei typische Beispiele für diese Klasse von Spuren.

Das erste Beispiel ist in Abbildung 1.5 dargestellt. Es handelt sich um Kratzspuren eines Werkzeugs (z.B. Stemmeisen, Schraubenzieher, Axt) auf einem verformbaren Untergrund. Durch die individuelle Verwendungsgeschichte eines Werkzeugs entstehen Verformungen, die für das einzelne Werkzeug jeweils charakteristisch sind. Durch diese Verformungen hinterlässt das Werkzeug bei der Verwendung charakteristische Spuren.

Als zweites Beispiel betrachten wir die Spuren, die bei der Verwendung von Schusswaffen an der Kugel entstehen. Der Lauf einer Schusswaffe enthält nämlich typischerweise ein spiralförmiges Rillenmuster (siehe Abbildung 1.6), das die Kugel vor dem Austritt aus dem Lauf in eine Rotation versetzt. Diese Rotation verleiht der Kugel eine deutlich bessere Flugstabilität. Da das Rillenmuster bei der Waffenherstellung mechanisch erzeugt wird und sich durch Abnutzung ändert, existieren auch bei baugleichen Waffen Unterschiede, die sich beim Schuss auf die Kugel übertragen.

Um eine am Tatort gefundene Kugel einer sichergestellten Waffe zuzuordnen, werden weitere Schüsse mit der Waffe abgegeben und die abgeschossenen

Kugeln mit der fraglichen Kugel vom Tatort verglichen. Hierbei kommt ein spezielles Vergleichsmikroskop zum Einsatz, das in Abbildung 1.7 schematisch dargestellt ist. Unter dem Mikroskop wird die Frage untersucht, ob es Übereinstimmungen zwischen den Linienmustern auf beiden Kugeln gibt. Etwaige Übereinstimmungen können dann anhand von Fotos dokumentiert werden (siehe Abbildung 1.8).

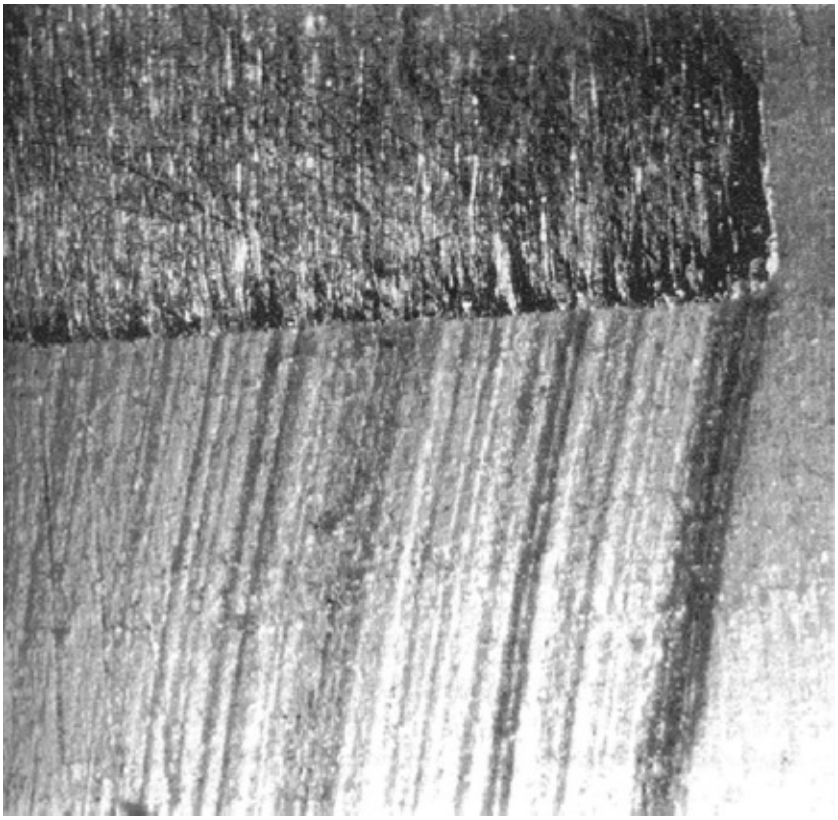


Abbildung 1.5: Kratzspuren eines Werkzeugs (oben) auf einem verformbaren Material (Inman u. Rudin, 2000, S. 97).

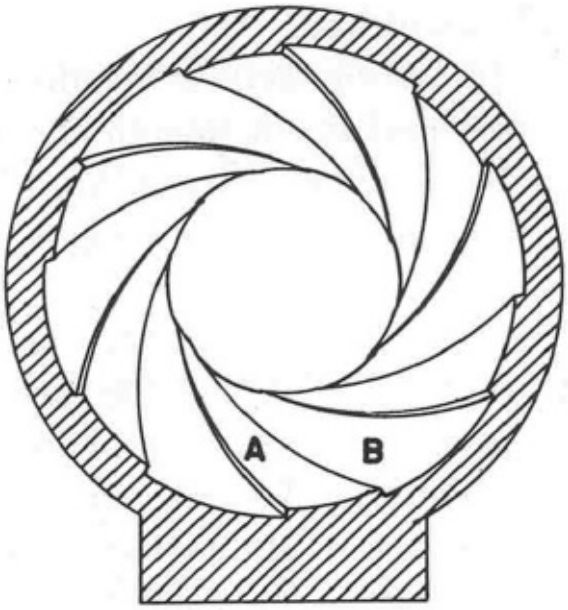


Abbildung 1.6: Schematische Darstellung des Laufs einer Schusswaffe (Kirk, 1974, S. 387).

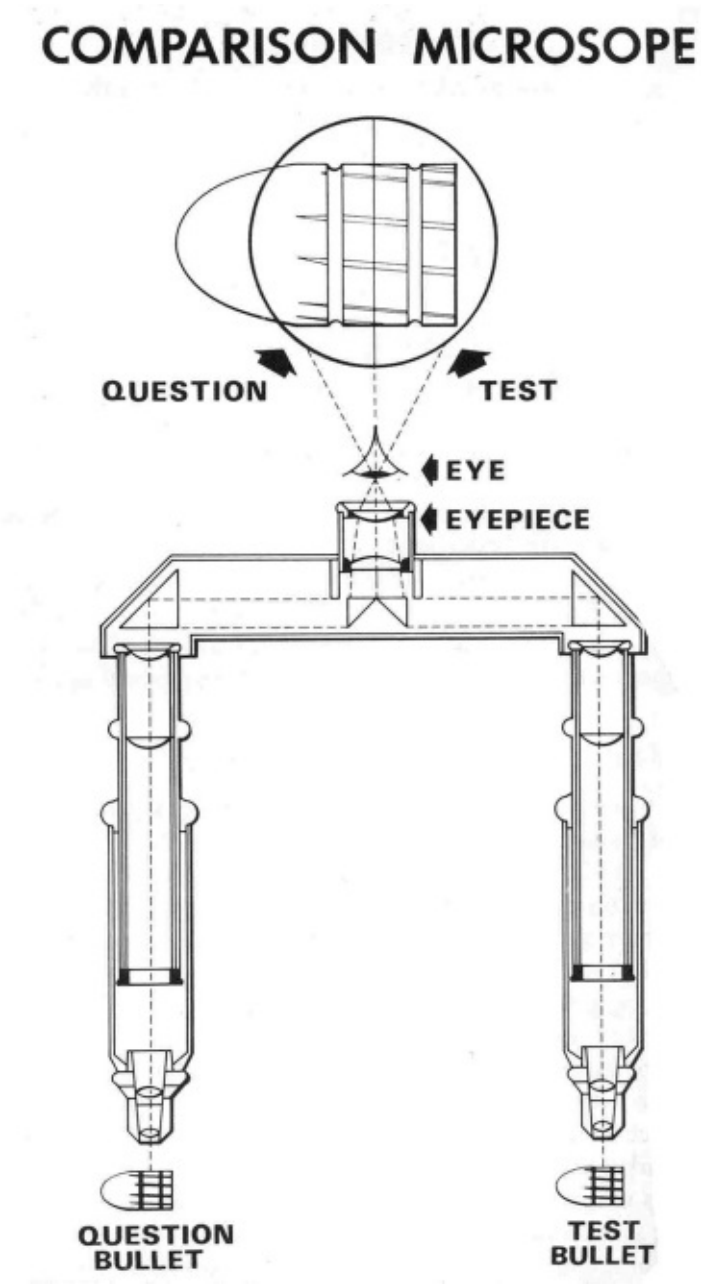


Abbildung 1.7: Schema eines Vergleichsmikroskops (Kirk, 1974, S. 364).

Ein besonderes Beispiel ist in Abbildung 1.9 dargestellt. Abgebildet ist die stark vergrößerte Draufsicht einer Kugel, auf der der Abdruck eines Webmusters erkennbar ist. Dieses Muster konnte dem Stoff einer Polizeiuniform zugeordnet werden. Außerdem wurden anhaftende Fasern des Uniformstoffes an dieser Kugel sichergestellt. Dies ist also ein Beispiel, bei dem sowohl die Übertragung von Materie als auch von Mustern stattgefunden hat.

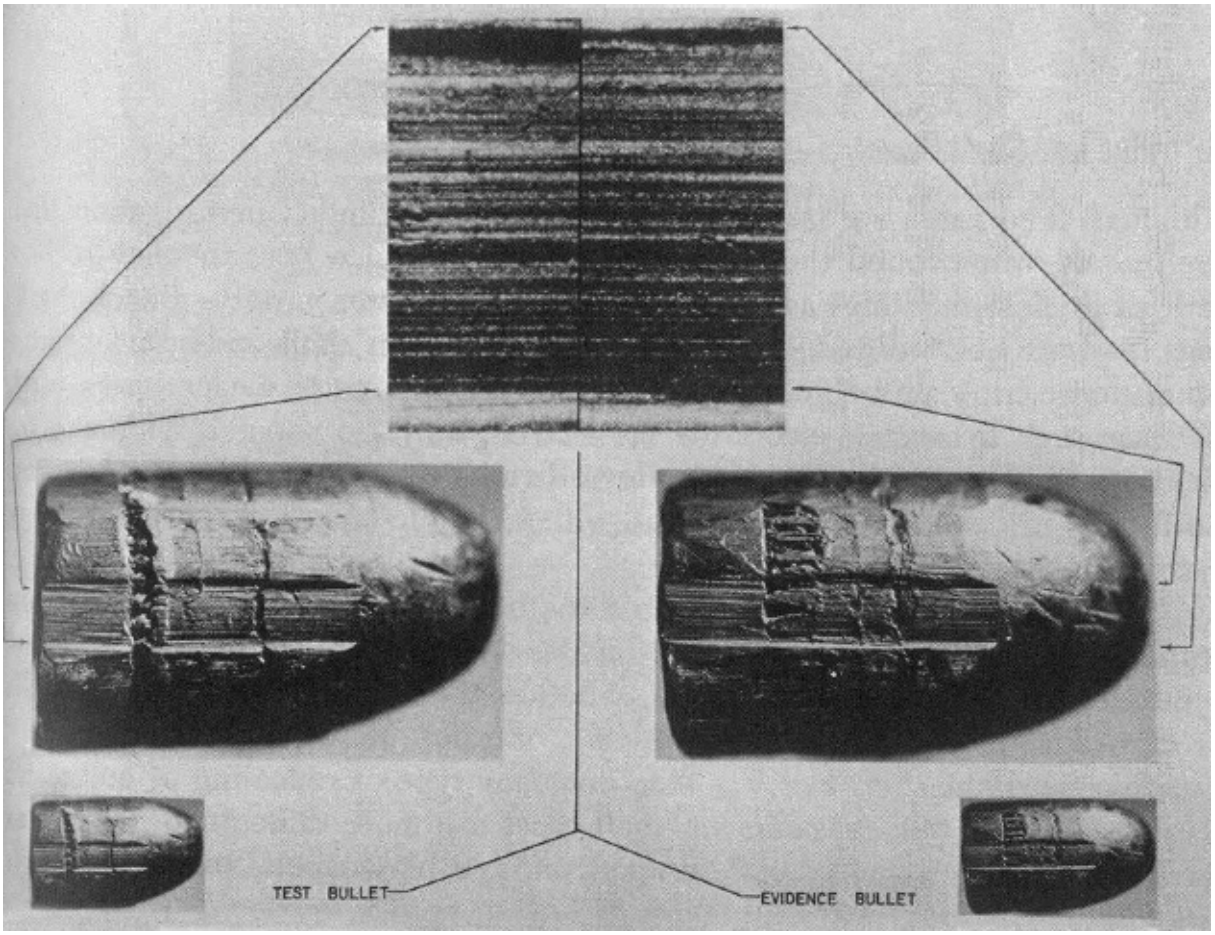


Abbildung 1.8: Vergleich von Spuren auf Kugeln (Kirk, 1974, S. 397).

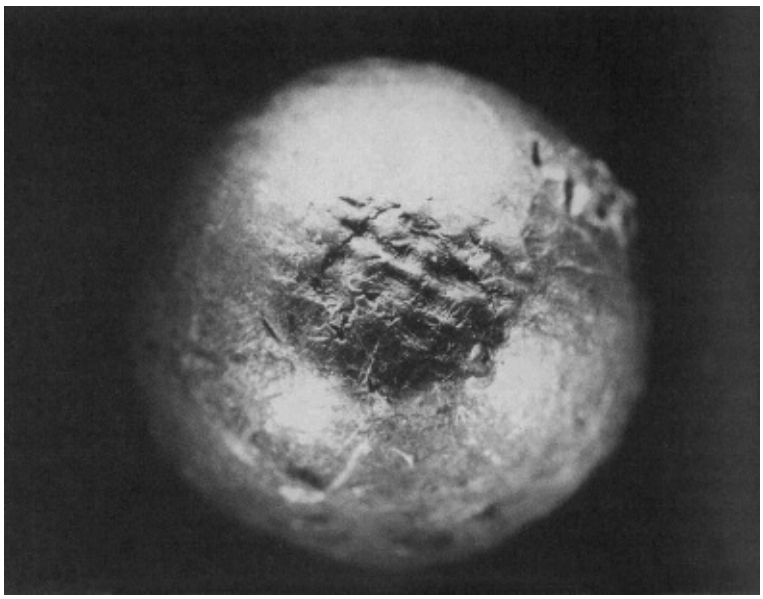


Abbildung 1.9: Beispiel für gleichzeitigen Austausch von Muster und Materie (Kirk, 1974, S. 122).

1.2.7 Theorie des Transfers

Zusammenfassend gibt es also zwei Formen des Transfers (Inman u. Rudin, 2000, Kapitel 4.D):

1. Übertragung von Materie (***physical transfer***). Hierbei geht man in der Regel davon aus, dass sich unter einer gewissen Energieeinwirkung ein Objekt zerteilt und Einzelteile davon von einer Quelle auf ein Ziel übertragen werden. Typischerweise fällt die Energie beim Kontakt an.
2. Übertragung von Mustern (***transfer of traits***). Hierbei werden charakteristische Formeigenschaften von einem Objekt auf ein anderes übertragen, ohne dass notwendigerweise Materie ausgetauscht wird.

Die Übertragung von Materie oder Mustern ist an einem Tatort in der realen Welt nie vollständig zu vermeiden. Wie zuvor bereits erwähnt, ist insbesondere die Verfügbarkeit von mikroskopischen Spuren in der Praxis sehr hoch (Kirk, 1974, S. 2). Ein uneingeschränkter Ermittler hat (jedenfalls theoretisch) die Möglichkeit, den Tatort aus unendlich vielen Perspektiven zu betrachten. Somit besteht in der realen Welt immer eine bestimmte Wahrscheinlichkeit, dass diese Spur auch entdeckt wird (Böhme u. a., 2009, S. 92f). In der realen Welt ist deshalb eine Tat ohne Spuren (***perfektes Verbrechen***) theoretisch ausgeschlossen.

1.3 Rekonstruktion eines Tathergangs

Ziel des Ermittlers ist immer die Rekonstruktion des ***wahren*** Tathergangs unter Einbeziehung der aufgefundenen Spuren. Praktisch werden jeweils Hypothesen über eine Menge von möglichen Tathergängen aufgestellt, die den aufgefundenen Spuren nicht widersprechen. Gemäß der zuvor geschilderten Theorie nimmt man an, dass alle Spuren ausschließlich auf den Prinzipien der Zerteilbarkeit und des Transfers beruhen. Spuren sind dann jeweils Hinweise auf einen ***Kontakt*** zwischen zwei Objekten.

1.3.1 Ereignisse

Wenn festgestellt wurde, dass zwei Objekte in Kontakt waren, spricht man von einem ***Ereignis*** (***event***). ***Rekonstruktion*** bedeutet nun, alle ermittelten Ereignisse in einen räumlichen und zeitlichen Zusammenhang zu bringen.

Beispiele für Ereignisse sind etwa:

- Person ***P*** war am Ort ***X*** (etwa weil Fingerabdrücke von ***P*** am Ort ***X*** gefunden wurden).
- Ein Schuh ***S*** war am Ort ***Y*** (etwa weil ***S*** einen charakteristischen Sohlenabdruck an Ort ***Y*** hinterlassen hat).
- Ein Auto ***A*** war am selben Ort, an dem auch Person ***Q*** war (etwa weil man Blutspuren von ***Q*** auf der Motorhaube von ***A*** gefunden hat).

Alle diese Ereignisse müssen individuell hergeleitet und begründet werden. Die Basis dafür ist der Vorgang der Assoziation.

1.3.2 Assoziation

Assoziation bezeichnet den Vorgang, bei dem der Kontakt zwischen zwei Objekten festgestellt wird. Als Ergebnis der Assoziation steht ein Ereignis. Als Beispiele für eine Assoziation können genannt werden:

- Die Zuordnung eines konkreten Schuhs zu einem konkreten Schuhabdruck.
- Die Zuordnung einer konkreten Kugel zu einer konkreten Waffe.
- Die Zuordnung einer konkreten Blutspur zu einer konkreten Person.
- Die Zuordnung eines konkreten Fingerabdrucks zu einer konkreten Person.

Ein wesentlicher Bestandteil der Assoziation ist die Quantifizierung der Irrtumswahrscheinlichkeit. In den klassischen forensischen Wissenschaften wird die Quantifizierung von Assoziationsaussagen unterschiedlich gehandhabt. Häufig haben sich in der Praxis Richtwerte herausgebildet, die allgemein als Nachweis einer Assoziation akzeptiert werden (ohne exakte Quantifizierung). Ein Beispiel ist der Bereich der Daktyloskopie. Hier werden Fingerabdrücke auf bestimmte Merkmale hin untersucht (beispielsweise das Zusammenlaufen oder Auseinandergehen von Hautrillen an bestimmten Punkten des Fingers). Fingerabdrücke werden als übereinstimmend angesehen, wenn es eine

bestimmte Mindestanzahl (z. B. acht oder zwölf) übereinstimmender Merkmale und keine widersprüchlichen Merkmale gibt. Eine echte Quantifizierung hingegen würde eine Aussage machen, bei wie vielen Personen eine spezifische Kombination von Merkmalen vorkommt. Weisen sowohl die Abdrücke am Tatort als auch die Abdrücke eines Verdächtigen diese Merkmalskombination auf, dann kann man die Irrtumswahrscheinlichkeit der Assoziation bestimmen. Weist beispielsweise eine Person in 10.000 Personen die Merkmalskombination auf, dann beträgt die Irrtumswahrscheinlichkeit 1 : 10.000. Repräsentative Studien über die Verteilung von Merkmalen in der Bevölkerung sind jedoch rar.

Besonders gut untersucht sind Merkmalsverteilungen und entsprechende Wahrscheinlichkeiten im Bereich der DNA-Analyse. Hier gibt es relativ viele und umfassende Studien über die Verteilung von bestimmten Merkmalen in der menschlichen Erbsubstanz. Liegt genügend DNA-Material vor, kann man Wahrscheinlichkeiten berechnen, die eine bestimmte Person exakt identifizieren (beispielsweise bei einer Irrtumswahrscheinlichkeit von 1 : 25 Milliarden) (Benecke, 2001, S. 7f). Es besteht aber auch bei der DNA-Analyse die Gefahr einer fälschlichen Assoziation, nämlich durch verunreinigte, vertauschte oder falsch beschriftete Spuren.

Bei vielen Fällen in der Praxis ist es schwierig, die Irrtumswahrscheinlichkeit der Assoziation anzugeben. Trotzdem sollte bei jeder Assoziation hinterfragt werden, auf welcher Basis der Schluss erfolgte.

1.3.3 Der Weg zur Assoziation

Die Assoziation und das daraus resultierende Ereignis sind nur das Endergebnis einer Folge von Schritten, die nach und nach die Mengen der zueinander passenden Objekte derart einschränken, dass am Ende nur zwei Objekte übrig bleiben. In der Literatur (Inman u. Rudin, 2000, Kapitel 6) werden die folgenden Schritte definiert: Identifizierung, anschließend Klassifizierung und Individualisierung einer Spur.

Bei der ***Identifizierung*** (***identification***) wird die prinzipielle Tauglichkeit der Spur als Beweismittel geprüft. Es wird gefragt: ***Was ist es?*** Beispielsweise muss eine Blutspur zunächst gefunden und dann als Blutspur erkannt werden. Identifikation kann charakterisiert werden als die Einschränkung der potentiellen Spuren am Tatort „mit bloßem Auge“.

Bei der ***Klassifizierung*** einer Spur (***classification***) wird die Menge der Spuren weiter eingegrenzt durch eine genauere Analyse der Form, der Größe, des

Gewichts, der Temperatur oder der Oberflächenstruktur des Gegenstandes. Hier wird gefragt: ***Zu welcher Klasse von Gegenständen gehört das Objekt?*** Charakteristisch für die Klassifizierung ist die Tatsache, dass man spezielle Werkzeuge wie eine Waage, ein Längenmaß, ein Mikroskop oder einen chemischen Test benötigt. Beispielsweise kann die Herkunft eines am Tatort gefundenen länglichen Gegenstandes zunächst als biologisch klassifiziert werden. Weitere Untersuchungen erlauben dann die genauere Klassifizierung als Haar. Anschließend kann man die Spur als menschliches Haar klassifizieren. Klassifizierungsmerkmale entstehen in der Regel aus ***kontrollierten*** Herstellungsprozessen, die dann jeweils charakteristisch für die Klasse von Objekten sind.

Mit ***Individualisierung*** (***individualization***) bezeichnet man die Zuordnung der Spur zu einer eng umgrenzten Menge von Objekten, die potentiell die Spur verursacht haben könnten. (Der Begriff wird oft fälschlicherweise verengt als die Zuordnung einer Spur zu einem Individuum verstanden.) Idealerweise führt die Individualisierung zu einer 1:1-Zuordnung zwischen Spur und dem Referenzobjekt, das die Spur verursacht hat. Die Merkmale, die eine Individualisierung erlauben, entstammen in der Regel ***zufälligen*** und ***unkontrollierten*** Prozessen, wie sie beispielsweise durch Abnutzung geschehen.

Wie eingangs geschildert, steht am Ende dieser Kette die Assoziation, also die möglichst genau quantifizierbare Feststellung des Kontakts zwischen zwei Objekten, beispielsweise die Zuordnung einer DNA-Spur am Tatort zu einer bestimmten Person.

Abbildung 1.10 fasst den Weg zur Assoziation nochmals zusammen. Im oberen Bereich der Abbildung ist der Tatort dargestellt. Am Tatort gibt es verschiedene Objekte, die zunächst als mögliche Spur identifiziert werden müssen. Im nun folgenden Prozess wird versucht, das Objekt am Tatort mit einer eng umgrenzten Menge an „passenden“ Objekten zu assoziieren. Dies geschieht durch eine schrittweise Eingrenzung der Objektmenge in der restlichen Welt mittels Klassifizierung und Individualisierung.

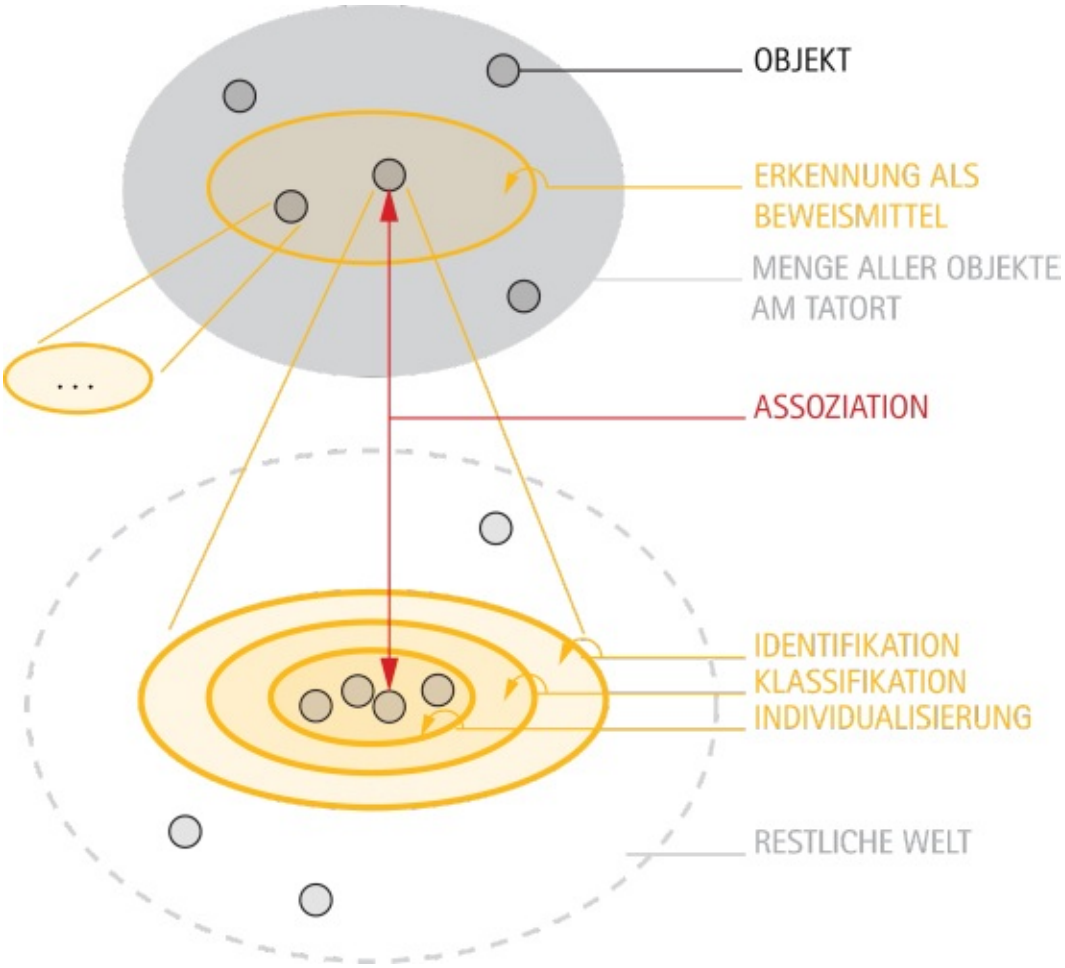


Abbildung 1.10: Schematische Darstellung des Weges zur Assoziation.

1.3.4 Beispiele

Wir betrachten im Folgenden einige Beispiele, um die Schritte von der Identifizierung bis zur Individualisierung zu illustrieren.

- Am Tatort existiert der Abdruck eines Objektes im Sand. Die Identifizierung geschieht durch Hinschauen: Es handelt sich um einen Schuhabdruck. Die Klassifizierung geschieht mittels zusätzlicher Messinstrumente wie einem Metermaß und einem Herstellerverzeichnis (das Logo des Herstellers ist im Abdruck erkennbar): Es handelt sich um einen Herrenschuh der Größe 44 einer bestimmten Marke. Die Individualisierung erfolgt durch die Beobachtung von bestimmten Unregelmäßigkeiten im Abdruck, etwa Risse in der Sohle oder charakteristisch abgenutzte Stellen.

- Am Tatort einer Schießerei wird ein kleines metallisches Objekt gefunden. Es wird durch bloßes Hinschauen als Kugel einer Schusswaffe und damit als potentielle Spur identifiziert. Anschließend wird die Kugel von ihrer Größe und Beschaffenheit her als Pistolenkugel eines gewissen Kalibers klassifiziert. Die Individualisierung erfolgt durch die Beobachtung von charakteristischen Kratzspuren an den Seiten der Kugel.

1.3.5 Notwendigkeit von Identifizierung und Klassifizierung

Unter bestimmten Voraussetzungen ist es gar nicht notwendig, die volle Wegstrecke bis hin zur Individualisierung zu gehen. Abhängig vom Delikt und dem damit verbundenen Straftatbestand kann man manchmal schon nach der Identifizierung oder der Klassifizierung aufhören. Ein Beispiel hierfür ist die Identifizierung oder Klassifizierung von Objekten wie Betäubungsmitteln, deren Besitz möglicherweise bereits unter Strafe steht. Es ist dann für die Ermittlung meist nicht mehr relevant, von welchem Drogendealer oder aus welchem Drogenlabor das Objekt genau stammt. Ein anderes Beispiel ist der Besitz (gedruckter) kinderpornographischer Schriften. Für die Ermittlung der Umstände einer Straftat mag vielleicht relevant sein, woher die Schriften bezogen wurden (Individualisierung). Für die Anklage kann allerdings bereits die Klassifizierung eines Dokuments ausreichen.

1.4 Zusammenfassung

In diesem Kapitel haben wir Grundbegriffe und Prinzipien der klassischen Forensik betrachtet. Implizit haben wir dabei vorausgesetzt, dass es sich bei Spuren immer um physische Spuren handelt. Im folgenden Kapitel treten wir in die digitale Welt des Cyberspace ein und untersuchen, inwiefern die Begriffe der klassischen Forensik auf digitale Spuren und die Informatik anwendbar sind.

Kapitel 2

Digitale Spuren

Autoren: Andreas Dewald, Felix Freiling

In diesem Kapitel definieren wir den Begriff der ***digitalen Spuren*** und stellen diesen in den Kontext der klassischen Forensik. Wir möchten damit zeigen, dass sich die Informatik ohne Weiteres in die Tradition der klassischen forensischen Wissenschaften stellen lässt. Zwar verlangt die Untersuchung digitaler Spuren manchmal neue Methoden, aber die Informatik ist keineswegs eine neue forensische Wissenschaft, für die andere Gesetze als bei den übrigen forensischen Wissenschaften gelten. Fast alle Prinzipien, die man in der klassischen Forensik für physische Spuren entwickelt hat, lassen sich auch auf digitale Spuren anwenden. Die vermeintlichen Unterschiede bei der Behandlung digitaler Spuren führen lediglich zu einer Betonung bestimmter forensischer Prinzipien, die bei der Betrachtung von physischen Spuren nur am Rande vorkamen.

2.1 Definition und Abgrenzung

Eine der gängigsten Definitionen digitaler Spuren stammt von Casey (2004, S. 12) und lautet wie folgt:

Definition 1 (digitale Spuren) ***Digitale Spuren (digital evidence) sind Spuren, die auf Daten basieren, welche in Computersystemen gespeichert oder übertragen worden sind***.

So genau diese Definition erscheint, so unscharf bleibt sie bei näherer Betrachtung. Beispielsweise wird nicht klar, worin die „digitale“ Natur der Spuren basiert. Basiert sie rein auf der Verarbeitung durch Computersysteme?

Falls ja, würden auch analog arbeitende Computer (Giloi u. Lauber, 1963) digitale Spuren erzeugen, was kontraintuitiv wäre. Basiert sie hingegen rein auf der diskreten (also digitalen) Codierung der Daten als Nullen und Einsen, dann wären auch handschriftliche Berechnung boolescher Ausdrücke digitale Spuren.

Exkurs 2 (Analogrechner) ***Bei Analogrechnern nutzt man aus, dass bestimmte mathematische Probleme durch physikalische Phänomene approximierbar sind. Ein einfacher mechanischer Analogrechner ist beispielsweise der*** Rechenschieber, ***bei dem man mit Hilfe von Verschiebebewegungen logarithmischer Skalen multiplizieren und dividieren kann. Es existieren auch Analogrechner auf elektronischer Grundlage, etwa zum Lösen von Differentialgleichungen***.

Um diese Situation etwas aufzuklären, kehren wir kurz zurück zur klassischen Forensik (Kapitel 1). Denn digitale Spuren sind zunächst immer auch physische Spuren, wie beispielsweise die Magnetisierung auf der Oberfläche einer Festplatte, elektromagnetische Wellen auf einem Datenkabel oder der Ladezustand von Speicherzellen im Hauptspeicher. Insofern können alle Begriffe und Prinzipien der klassischen Forensik direkt auf digitale Spuren angewendet werden, wie wir nun näher ausführen.

2.1.1 Information und Träger bei digitalen Spuren

In der klassischen Forensik haben wir zwischen ***Spureninformation*** und ***Spurenträger*** unterschieden: Der Spurenträger „trägt“ dabei die Spureninformation. Bei digitalen Spuren reduziert man die Spureninformation meist auf die ***diskrete Repräsentation*** der gemäß eines bestimmten Codierungsschemas auf dem Spurenträger gespeicherten Daten. Das Codierungsschema (also wann man eine Eins und wann man eine Null liest) ist dabei meist im Spurenträger gleich „mitverbaut“, also in der Festplatte oder dem RAM-Chip. Durch diese Abstraktion wird die Bindung zwischen Spurenträger und Spureninformation extrem zerbrechlich: Die Spureninformation kann nämlich theoretisch auf einem beliebigen Spurenträger gespeichert werden. Dies korrespondiert mit der Beobachtung, dass man in der digitalen Welt Daten „perfekt kopieren“ kann, also ohne die Möglichkeit, Original und Kopie später unterscheiden zu können.

Die schwache Verbindung von Spurenträger und Spureninformation im Falle von digitalen Spuren verdeutlicht den Unterschied zwischen beiden Konzepten: Der Spurenträger ist das „Speichermedium“, die Spureninformation ist die auf dem Speichermedium „gespeicherte“ Information. Bei der Reduktion auf digital (also als Bits) codierte Informationen abstrahiert man von vielen weiteren Informationen, die der Spurenträger enthält.

Beispiel 2 (Informationsverlust durch digitale Betrachtung) ***Digitale Datenübertragungen auf einem Kabel basieren auf Spannungsänderungen. So kann beispielsweise eine Spannung von 0 V eine Null repräsentieren und eine Spannung von 5 V eine Eins. Beim Anlegen der Spannung schwankt diese üblicherweise etwas, so dass bestimmte Spannungspegel angesetzt werden (etwa 0,5 V und 4,5 V), „ab denen“ der auf dem Kabel angelegte Spannungswert als Null oder Eins interpretiert wird. Der wirklich übertragene Wert (also die genaue Spannung) geht bei dieser Interpretation verloren. Ähnlich verhält es sich bei der Speicherung und Interpretation von Daten, die auf Festplatten oder im RAM-Speicher eines Rechners gespeichert werden. Die Interpretation beruht also auf Codierungsannahmen. Die resultierenden Daten jedoch sind (bei zuverlässiger Speicherung) exakt dieselben, die vorher gespeichert wurden***.

Der Fokus auf die reine Spureninformation unter Vernachlässigung des Spurenträger ist aber keine spezielle Eigenart digitaler Spuren. Solange klar ist, welche Information auf dem Spurenträger wichtig ist, kann man diese Spureninformation natürlich auch auf einem anderen Spurenträger ablegen. Wenn vom Spurenträger „blutiger Handschuh“ lediglich die Täter-DNA relevant ist, dann kann man die Ergebnisse der DNA-Analyse in einem Bericht niederlegen. Die Spureninformation wechselt dann den Spurenträger: vom blutigen Handschuh auf ein Blatt Papier.

2.1.2 Integrität

Die Betonung der Spureninformation, wie sie digitalen Spuren eigen zu sein scheint, hat auch noch andere Konsequenzen. Zunächst müssen Begriffe wie Integrität und Authentizität neu betrachtet werden. Bezüglich Integrität ist die Frage relevant, ob ein digitales Objekt (beispielsweise eine Datei auf einem Datenträger) seit der Sicherung verändert worden ist. Bei einer engen

Verbindung zwischen Spureninformation und Spurenträger kann man den Spurenträger selbst untersuchen und diesen aus weiteren Perspektiven betrachten, etwa die Tinte eines Briefes, die Handschrift auf einem Zettel, oder das Alter des Papiers. Bei der reinen Betrachtung der Spureninformation kann man auch eine Plausibilitätsprüfung des Inhalts durchführen und beispielsweise vergleichen, ob die Informationen widersprüchlich sind zu den mit der Spur verbundenen Behauptungen. Falls die Spur beispielsweise 2015 gesichert wurde, wäre es merkwürdig, wenn sich die Spureninformationen auf Ereignisse aus 2016 beziehen. Diese Form von Integritätsprüfung korrespondiert mit dem Begriff der ***semantischen Integrität*** aus der IT-Sicherheit (Gollmann, 2011).

Auch für die allgemeineren Begriffe Authentizität und Integrität bietet die IT-Sicherheit etablierte Definitionen an (Gollmann, 2011). Dabei wird oft übersehen, dass die Prüfung von Authentizität und Integrität stark auf Annahmen basiert.

Beispiel 3 (Integrität mittels Hashwerte) ***Eine*** Hashfunktion ***ist eine mathematische Funktion, die eine beliebig lange Bitfolge m auf einen Wert h mit fester Länge abbildet. Der Wert h wird häufig als*** Hashwert (***hash, digest) von m bezeichnet. Der Hashwert h hat dabei die besondere Eigenschaft, „charakteristisch“ für m zu sein (für einen genaueren Überblick siehe Menezes u. a. (1997) oder Kapitel 6***).

Hashwerte besitzen viele Eigenschaften von Fingerabdrücken: ***Intuitiv kann man den Hashwert dazu verwenden, um m zu identifizieren. Auch kann man sich den Hashwert nicht aussuchen, er ist der Bitfolge m quasi angeboren***.

Wenn man also die Integrität einer Bitfolge m prüfen will, so kann man den Hashwert h von m berechnen und diesen vergleichen mit einem andernorts hinterlegten Hashwert h. Wenn h und h′ übereinstimmen, dann liegt der Berechnung von h derselbe Bitstring zugrunde, der auch der Berechnung von h′ zugrunde lag. Seit der Berechnung von h′ hat sich also m nicht verändert.

Die eben im Beispiel beschriebene technische Auffassung von Integrität beruht auf der Annahme, dass der hinterlegte Hashwert verlässlich und korrekt berechnet wurde, beispielsweise zum Zeitpunkt der Sicherung der originalen Bitfolge, und dass der Hashwert selbst authentisch ist und nicht verändert worden ist (Lynch, 2000).

2.1.3 Authentizität

Authentizität ist bei digitalen Spuren noch etwas schwieriger zu definieren als bei physischen Spuren. Im Prinzip bedeutet Authentizität lediglich, dass die vorgelegte Spur tatsächlich die Spur ist, die durch die mit ihr verbundenen Behauptungen beschrieben wird. Bei physischen Spuren gibt es über den Spurenträger sehr viele Möglichkeiten, um diese Behauptungen anhand des Objekts selbst zu prüfen, etwa die Behauptung, dass es sich bei dem Objekt um einen „blutigen Handschuh“ handelt. Bei digitalen Spuren verwendet man häufig den aus der IT-Sicherheit etablierten Begriff der Authentizität (Gollmann, 2011), wie er durch digitale Signaturen realisiert werden kann.

Beispiel 4 (Authentizität durch digital signierten Hashwert) ***Um Nachrichten digital zu signieren, kann man ein asymetrisches Verschlüsselungsverfahren verwenden. Bei einem asymetrischen Verschlüsselungsverfahren besitzt jede Person einen öffentlichen Schlüssel und einen privaten Schlüssel. Der öffentliche Schlüssel ist allen anderen Personen bekannt. Von den privaten Schlüsseln kennt jede Person nur den eigenen (mehr Details folgen in Kapitel 6, für den Überblick siehe auch Menezes u. a. (1997)). Mit dem privaten Schlüssel kann man Nachrichten signieren. Die Eigenschaften dieser Operation sind analog zum Setzen der eigenen Unterschrift unter die Nachricht. Jede andere Person kann prüfen, ob die Unterschrift von der angegebenen Person stammt, und auch nur diese Person kann die Unterschrift geleistet haben. Aus Effizienzgründen signiert man jedoch meist nur den Hashwert der Nachricht. Man kann folglich eine digitale Spur als authentisch ansehen, wenn wir einen korrekten Hashwert besitzen, der von der Person digital signiert wurde, die die Spurensicherung durchgeführt hat***.

Auch bei der eben geschilderten Prüfung der Authentizität machen wir viele Annahmen. Zuvorderst vertrauen wir in die Verlässlichkeit der Person, die die Sicherung (mitsamt digitaler Signatur) durchgeführt hat. Das bedeutet vor allem auch, dass nur diese Person ihren privaten Schlüssel kennt und niemand anderes. Auch vertrauen wir in die mathematische Sicherheit des digitalen Signaturverfahrens. Unter der Annahme, dass das Signaturverfahren nicht gebrochen worden ist, wissen wir dann lediglich, dass jemand die Unterschrift geleistet hat, der den zu einer Person gehördenden privaten Schlüssel kennt — nicht mehr und nicht weniger.

Digitale Signaturen sind also keineswegs mit handgeschriebenen Unterschriften zu vergleichen. Eine handgeschriebene Unterschrift kann viele

verschiedene Bedeutungen haben (Lynch, 2000): Mit einer Unterschrift kann man anzeigen,

- dass man der Autor des Dokumentes ist,
- dass man das Dokument oder einen Teil des Dokumentes zur Kenntnis genommen hat,
- dass man das Dokument erhalten hat, oder
- dass man dem Inhalt des Dokumentes zustimmt.

Bei Unterschriften geht es also nicht immer nur um ihre Bedeutung sondern auch um den Wirkungsbereich, auf den sie sich beziehen. Dies ist schwer mit digitalen Signaturen nachzubilden, vor allem, wenn sich Dokumente über die Zeit verändern (zum Beispiel durch Umformatierung).

Exkurs 3 (Urkundenwesen) ***Der Übergang von physischen Dokumenten zu rein digitalen Archiven wird seit langem im Bereich der Buchwissenschaften diskutiert, speziell im*** Urkundenwesen (***diplomatics) (Duranti, 1998). Dort spielen Authentizität und Integrität von Dokumenten eine entscheidende Rolle (Lynch, 2000), etwa bei Besitzurkunden oder bei Verträgen. Die Bezüge zur digitalen Forensik sind bereits lange etabliert und fruchtbar (Dardick u.a., 2014; Rogers, 2013, Sect. 3.8***).

2.2 Entstehung digitaler Spuren

Wir möchten nun untersuchen, inwieweit sich die in Kapitel 1 geschilderten grundlegenden Entstehungsprinzipien von Spuren in die digitale Welt übertragen lassen und widmen uns zunächst den Phänomenen der Zerteilbarkeit und des Transfers in der digitalen Welt. Anschließend befassen wir uns mit dem Konzept der Assoziation.

2.2.1 Zerteilbarkeit und Transfer in der digitalen Welt

Intuitiv kann die digitale Welt als Zustandsautomat beschrieben werden. Auch wenn dieser Automat in die reale Welt eingebettet ist, bleiben dessen Zustände diskret. In diesem Sinne gibt es in der digitalen Welt auch keine Materie: Alle dort manipulierten Artefakte sind schlussendlich Daten, die in diskreter

Kodierung im Speicher eines Rechners liegen. Eine Zerteilbarkeit von Materie (***divisibility***), die die Grundlage für den physischen Transfer (***transfer of matter***) in der realen Welt bildet, gibt es somit nicht im selben Maße (immerhin werden Ladungen und Magnetisierungspotentiale hinund hergeschoben). Es gibt in der digitalen Welt also keine Analogie für das Konzept des Austauschs im Sinne von Locards Austauschprinzip (siehe Kapitel 1).

Übertragung von Materie ist, wie wir ebenfalls aus Kapitel 1 wissen, jedoch nur ***eine*** Art von Transfer, die im Rahmen einer Handlung erfolgen kann. Die andere Art der Übertragung ist die Übertragung von Mustern (***transfer of traits***). Diese Art der Übertragung findet in der physischen Welt immer dann statt, wenn Information von einem Objekt zum anderen übertragen wird. Dieser Vorgang hat eine offensichtliche Analogie in der digitalen Welt, wo der Austausch von ***Informationen*** (Dateien, E-Mails etc.) eine so zentrale Rolle spielt.

Aber nicht nur zwischen Rechnern werden Informationen ausgetauscht, auch innerhalb eines einzelnen Rechner werden Informationen zwischen Objekten im Speicher ausgetauscht.

Beispiel 5 (Übertragungen in der digitalen Welt) ***Beispiele für solche Übertragungen in der digitalen Welt sind etwa folgende Operationen:***

- ***Maschineninstruktion im Prozessor, zum Beispiel:*** `mov eax, ebx`
- ***Zuweisung in einer Programmiersprache, zum Beispiel:*** `x := y`
- ***Kopieroperation auf einem Computer, zum Beispiel in einem Kommandofenster (Shell):*** `copy file1. txt file2. txt`
- ***Kopieroperationen über das Netz, zum Beispiel in einem Kommandofenster:*** `scp file1. txt user@host. de:/`

Im Prinzip ist jede Datenverarbeitung Musterübertragung: Wenn ein Datum aus einem Teil des Speichers geladen, mit anderen Informationen verknüpft und anschließend wieder im Speicher abgelegt wird, dann drücken sich die Muster der Eingangsdaten im Muster der Ausgangsdaten ab. Bei Kopieroperationen ist direkt intuitiv ersichtlich, wie sich Muster von der Quelle zum Ziel hin verlagern. Betrachtet man jedoch auch den Gesamtzustand des Rechnersystems, dann bleiben viele Speicherbereiche (im RAM und auf der Festplatte) von der aktuellen Operation unberührt. Auch hier pflanzen sich Muster im Verlauf der Zeit fort. Wir werden später sehen, dass man auf Basis dieser

Musterübertragung, einen „Kontakt“ oder eine „gemeinsame Quelle“ zweier (digitaler) Objekte feststellen kann, genau wie in der klassischen Forensik auch.

2.2.2 Geschlossenes vs. offenes System

Die Beschränkung auf die (digital codierte) Spureninformation birgt eine weitere Problematik, die uns im Verlauf dieses Kapitels noch mehrfach begegnet, denn die Beschränkung ist fundamental: Mit der Abkehr vom Spurenträger schließt man unendlich viele Blickrichtungen aus, die weitere Spureninformationen tragen könnten. Dies kann fatale Folgen haben. Hätte man beispielsweise schon in den 1960er Jahren alle Spuren „digitalisiert“ und die Spurenträger vernichtet, dann wären alle DNA-Spuren verloren gegangen, weil die Bedeutung dieser Spurenart zu der Zeit noch unbekannt war.

Die Betrachtung digitaler Spuren verleitet dazu, ausschließlich in digitalen Dimensionen zu denken und die Bedeutung und die Einflüsse der physischen Welt auf diese Spuren zu vernachlässigen. Die digitale Welt/der Cyberspace wird dadurch als ***geschlossenes System*** wahrgenommen. Die implizite Annahme lautet dabei, dass der Austausch zwischen digitaler Welt und nicht-digitaler Welt nur über wohldefinierte Schnittstellen geschieht. Der Blick auf die digitale Welt und auch auf die Spuren darin ist jedoch dadurch stark eingeschränkt. Es gibt aber deutlich mehr Übergänge in die digitale Welt als nur die Tastatur des Computers. Wie wir vor allem im Bereich der Multimediaforensik (Kapitel 4) sehen werden, schlagen sich physische Phänomene direkt in digitalen Spuren nieder, die mit entsprechenden Techniken ausgewertet werden können. Notwendig dafür ist lediglich eine gewisse Grundkomplexität des digitalen Raumes, die jedoch heute problemlos auch von kleinsten Computern erreicht wird.

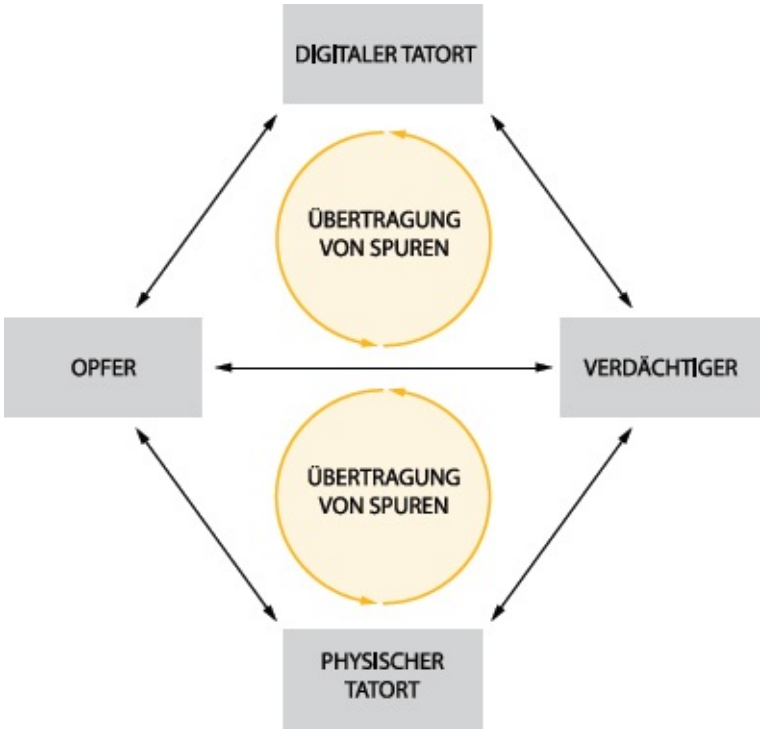


Abbildung 2.1: Die Erweiterung von Locards Austauschprinzip auf den digitalen Tatort nach Casey (2011, Abb. 1.1).

Konsequenterweise erweitert Casey (2011, S. 17) das ursprüngliche Austauschprinzip von Locard, das nur eine ***physical crime scene*** vorsah, um eine ***digital crime scene***, in dem ein Austausch von digitalen Spuren stattfinden kann (siehe Abbildung 2.1). Diese Gegenüberstellung verdeutlicht aber auch, dass ein digitaler Austausch von Spuren einhergehen kann mit einem physischen Austausch von Spuren, etwa wenn ein Schalter oder eine Taste physisch betätigt und dadurch Software gestartet wird, oder wenn (wie im Beispiel der Multimediaforensik) Umgebungseinflüsse durch entsprechende Sensorik die Spurenentstehung beeinflusst.

2.2.3 Abstraktionsschichten

Ein relevanter Aspekt digitaler Spuren beruht auf der Tatsache, dass digitale Spuren meist in einer für Menschen nicht direkt zugänglichen Form vorliegen. Wie auch bei vielen Arten von physischen Spuren, die man oft nicht mehr mit bloßem Auge erkennen kann, müssen digitale Spuren in der Regel immer zunächst extrahiert und in eine lesbare Form übersetzt werden. Diese

Aufbereitung erfordert den Einsatz von Werkzeugen, die eine Abstraktion bzw. Interpretation der physischen Spuren zeigen. Charakteristisch für moderne Computersysteme ist zudem, dass es meist mehrere Abstraktionsebenen gibt, auf denen die Daten dargestellt werden können.

Beispiel 6 (Abstraktionsschichten digitaler Spuren) ***Als Beispiel betrachten wir die Abstraktionsschichten, die durchlaufen werden, wenn eine E-Mail-Nachricht als digitale Spur untersucht wird:***

1. ***Interpretation der Magnetisierung der Festplatte (Bits***)
2. ***Interpretation der Bits durch eine Zeichenkodierung***
3. ***Interpretation der Zeichen durch ein Dateisystem***
4. ***Interpretation der Daten im Dateisystem als zusammengehörige Datei***
5. ***Interpretation der Datei als E-Mail***

Jede einzelne dieser Abstraktionsebenen kann Quelle für Interpretationsfehler sein. Daher ist es bei einer forensischen Untersuchung digitaler Spuren insbesondere erforderlich, die Plausibilität jeder einzelnen Interpretation bzw. Abstraktion zu prüfen.

2.3 Eigenschaften digitaler Spuren

In diesem Abschnitt betrachten wir verschiedene Eigenschaften digitaler Spuren, nach denen man diese klassifizieren kann. Viele dieser Eigenschaften haben Parallelen zu Eigenschaften von Spuren der klassischen Forensik.

2.3.1 Flüchtigkeit

Die Klassifizierung nach Flüchtigkeit ist aus Sicht der digitalen Forensik nützlich, da wie in der klassischen Forensik Spuren je nach Flüchtigkeit unterschiedliche Vorgehensweisen und Techniken zur Sicherung und Analyse erfordern. Flüchtigkeit bezieht sich einerseits auf den Spurenträger (d.h. die Art und Weise, wie Daten abgespeichert werden), andererseits auch auf die Geschwindigkeit der Datenverarbeitung. Generell unterscheiden wir zwei bzw. drei Arten der Flüchtigkeit. Zunächst gibt es den Unterschied zwischen

persistenten und flüchtigen Spuren. Flüchtige Spuren können dann nochmals unterteilt werden.

Persistente Spuren sind Spuren, die über einen vergleichsweise großen Zeitraum ohne Stromzufuhr erhalten bleiben. Persistente Spuren entstehen typischerweise auf Festplatten, CD/DVDs, Solid-State-Speichern (USB-Sticks, CF-/SD-Karten etc.) und Magnetbändern. Derartige Spuren können nahezu beliebig lange nach der Tat analysiert werden, ohne Gefahr zu laufen, dass Daten mit der Zeit verloren gehen.

Flüchtige Spuren benötigen eine Stromzufuhr, um erhalten zu bleiben. Wir unterscheiden diese Klasse von Spuren noch zusätzlich in ***flüchtige Spuren im weiteren Sinne und im engeren Sinne***. Im weiteren Sinne sind Spuren flüchtig, wenn sie nur mit einer entsprechenden Stromzufuhr dauerhaft gespeichert werden. Manchmal werden diese Spuren darum auch als ***semi-persistent*** bezeichnet. Bei unterbrochener Stromzufuhr bleiben sie nur kurze Zeit erhalten. Typischer Vertreter dieser Klasse von Spuren sind Daten im Hauptspeicher (RAM) eines Rechners. Diese Spuren müssen entweder im laufenden Betrieb analysiert werden oder es muss zur späteren Analyse eine Kopie der Daten auf einem persistenten Datenträger erstellt werden. Im engeren Sinne flüchtig sind Spuren, die im laufenden Betrieb und ***trotz dauerhafter Stromzufuhr*** nur temporär vorhanden sind. Beispiele sind Netzwerkdaten oder die Inhalte von Prozessorregistern. Diese Spuren müssen entweder im laufenden Betrieb analysiert werden, oder sie werden aufgezeichnet und auf einem persistenten Datenträger gespeichert. Beispielsweise kann durch das Mitschneiden von Netzwerkverkehr eine Protokolldatei der transportierten Daten über einen gewissen Zeitraum angefertigt werden.

2.3.2 Technische Vermeidbarkeit

Bei der Betrachtung von Spuren in Dateisystemen beobachtete Carrier (2005), dass die Veränderungen bestimmter Daten dazu führen können, dass das System nicht mehr richtig funktioniert. Beispiele hierfür sind Verweise auf Datenblöcke im Dateisystem. Ohne derartige Verweise wären die Inhalte einer Datei nicht auffindbar. Carrier (2005) bezeichnet diese Daten als ***essentiell*** (***essential data***). Andere Daten, wie etwa der eigentliche Dateiinhalt, können eine nahezu beliebige Form haben, ohne dass dies das Systemverhalten entscheidend beeinflussen würde. Sollen unter Benutzung des Systems Spuren manipuliert werden, so erzeugt die Veränderung essentieller Daten einen hohen Aufwand,

weil die Manipulationen unter Umständen zunächst wieder rückgängig gemacht werden müssen, bevor das System benutzt werden kann. Wir können also festhalten, dass die Manipulation essentieller Daten mit einem deutlich höheren Aufwand verbunden ist als die Manipulation nicht-essentieller Daten. Diese Beobachtung kann bei der Einschätzung des Beweiswertes gesicherter Spuren helfen.

Aufbauend auf Carrier (2005) formalisierten Freiling u. Gruhn (2015) den Begriff der essentiellen Daten und entdeckten zwei Varianten in Bezug auf eine vom System bereitzustellende Funktionalität (beispielsweise Dateien abzuspeichern):

- ***Strikt essentielle Daten*** (***strictly essential data***) sind für alle betrachteten Systeme essentiell, sie sind also der Kern dessen, was die Funktionalität ausmacht. In Dateisystemen sind beispielsweise die Blockverweise strikt essentielle Daten.
- ***Partiell essentielle Daten*** (***partially essential data***) sind nur für manche Systeme essentiell, für andere nicht. Beispiele für partiell essentielle Daten werden im Kapitel über Partitionssysteme (Kapitel 6) erläutert.

Man kann den Begriff der (strikt/partiell) essentiellen Daten sogar noch konzeptionell verallgemeinern und unterscheiden zwischen technisch vermeidbaren und technisch unvermeidbare Spuren.

Definition 2 (Technisch vermeidbare Spuren) ***Technisch vermeidbare Spuren sind Spuren, die um ihrer selbst Willen erzeugt wurden***.

Technisch vermeidbare Spuren sind nicht kritisch für das Funktionieren des Systems. Die Entstehung solcher Spuren kann lokal in der Regel durch eine Änderung der Systemkonfiguration eingeschränkt oder gänzlich unterbunden werden. Ein Beispiel für derartige Spuren sind Log-Dateien. Auch Daten, die im Rahmen einer Vorratsdatenspeicherung anfallen, gehören in diese Kategorie, da sie um ihrer selbst Willen erzeugt werden – auch wenn der Benutzer selbst nicht in der Lage ist, ihre Erzeugung zu verhindern.

Definition 3 (Technisch unvermeidbare Spuren) ***Technisch unvermeidbare Spuren sind Spuren, die unweigerlich anfallen und daher nicht durch***

einfache Änderungen an der Konfiguration eines Systems vermieden werden können.

Eine Veränderung technisch unvermeidbarer Spuren ist in der Regel nicht durch die Bordmittel des Betriebssystems erreichbar. Beispiele für derartige Spuren sind gelöschte Dateien im Dateisystem, alte Stackframes im Hauptspeicher oder Inhalte des DNS-Caches.

Aus Sicht des Ermittlers sind technisch unvermeidbare Spuren besonders interessant, da ihnen in der Regel ein höherer Beweiswert zugerechnet werden kann. Dies liegt daran, dass es deutlich aufwändiger ist, technisch unvermeidbare Spuren zu manipulieren als technisch vermeidbare Spuren. Insofern existiert eine Analogie zwischen technisch unvermeidbaren digitalen Spuren und mikroskopischen physischen Spuren. Technisch vermeidbare Spuren liegen zumeist als Inhaltsdaten in Form von Dateien vor. Sie gehören demnach meist zu den nicht-essentiellen Daten im Dateisystem und gelten allein dadurch bereits als weniger glaubwürdig. Technisch unvermeidbare Spuren benötigen aber eine besondere Expertise bei der Sicherung und Analyse.

2.3.3 Manipulierbarkeit

Wenn man die Betrachtung digitaler Spuren auf die Spureninformation reduziert, dann können digitale Spuren prinzipiell leicht manipuliert werden. Dies kann sowohl absichtlich als auch unabsichtlich geschehen. Die Manipulation digitaler Spuren hinterlässt auf der Ebene der digitalen Spureninformation nicht notwendigerweise sichtbaren Zeichen.

Beispiel 7 (Manipulation digitaler Spuren) ***Als Beispiel für die Manipulation digitaler Spuren betrachten wir einen Rechner, der mittels einer Live-CD bootet. Die Live-CD greift nicht auf die Festplatte des Rechners zu, sondern operiert rein von einer RAM-Disk. Mit einem entsprechenden Werkzeug (Hexeditor) wird anschließend ein einzelnes Bit auf der Festplatte verändert. Daraufhin wird der Rechner heruntergefahren und die Live-CD aus dem Laufwerk entfernt***.

Diese nicht (direkt) nachvollziehbare Manipulation von Spuren ist im Bereich der klassischen Forensik eher untypisch, denn dort geht man davon aus, dass jede Handlung eine Spur hinterlässt. Es ist aber festzuhalten, dass eine Person,

die Spuren nichtnachvollziehbar manipulieren möchte, immer sehr viele Annahmen über die Umgebung machen muss, egal ob in der physischen oder der digitalen Welt, wie das folgende Beispiel verdeutlicht.

Beispiel 8 (Manipulation digitaler Spuren) ***Ein Rechner wird wiederum mit einer Live-CD gebootet. Nach dem Systemstart wird mittels einer VoIP-Software (zum Beispiel Skype) ein verschlüsseltes Telefonat mit einer anderen Person geführt. Die Kenndaten der anderen Person werden dabei im laufenden Betrieb eingegeben. Nach dem Telefonat wird der Rechner heruntergefahren und die Live-CD entfernt. Damit die Handlung unentdeckt bleibt, müssen folgende Annahmen getroffen werden:***

- ***Es wird lediglich die Festplatte des Rechners auf Spuren untersucht***.
- ***Die Systembenutzung selbst hinterlässt keine zusätzlichen digitalen Spuren (etwa im BIOS)***.
- ***Im Netzwerk fällen keinerlei Spuren an, bzw. die dortigen Spuren werden nicht untersucht***.
- ***Die Systembenutzung selbst (also die Tatsache, dass die Person den Rechner überhaupt benutzt) ist keine Spur (etwa, weil es sich um den persönlichen Rechner der Person handelt)***.
- ***Das Vorhandensein einer Live-CD mit vorkonfigurierter VoIP-Software ist selbst keine Spur***.

Beachten Sie, dass die letzten beiden Spuren keine digitalen Spuren sind.

Wie bereits weiter oben auf Seite → erwähnt, basiert die Meinung, dass es im Bereich der digitalen Spuren grundsätzlich nicht-nachvollziehbare Manipulation gibt, auf der Annahme, dass es sich bei dem System um ein ***geschlossenes System*** handelt. Wie das Beispiel eben zeigt, entstehen Spuren aber auch außerhalb des eigentlich benutzten Systems. In der Praxis werden Rechner in der Regel nicht isoliert genutzt, sondern treten in Verbindung mit anderen Rechnern. Es entstehen also dort und auch im Netz selbst Spuren (beim Internet-Service-Provider oder im Router), die untersucht werden können. Außerdem ist das digitale System immer in die physische Welt eingebettet, in der auch Spuren entstehen, etwa in Form von Fingerabdrücken auf der Tastatur.

Die perfekte Manipulierbarkeit digitaler Spuren ist also ähnlich problematisch wie die perfekte Manipulierbarkeit physischer Spuren. Wenn man hingegen ***ausschließlich*** digitale Spuren betrachtet (also die reine Spureninformation), dann ist eine perfekte Manipulation denkbar. Diese beruht dann aber auf einer Selbstbeschränkung des Ermittlers. Wenn, wie im zuvor beschriebenen Beispiel, mit einer Live-CD nur ein einzelnes Bit verändert wird, dann können zusätzliche digitale Spuren vermieden werden, weil dieses Bit eben nur zwei Zustände annehmen kann (und keine Zwischenzustände, die auf eine Manipulation hindeuten könnten). In der digitalen Welt gibt es nur endlich viele Perspektiven, aus denen ein Ermittler eine digitale Spuren betrachten kann – im Gegensatz zur physischen Welt.

Ähnlich verhält es sich mit einer besonderen Form der Manipulation: der Vernichtung bzw. Löschung von digitalen Spuren. Vor allem das selektive Löschen einzelner Dateien ist in der Praxis schwieriger als gedacht. Zum Beispiel sind Dateien, die mit den Bordmitteln des Betriebssystems gelöscht werden, in der Regel noch lange Zeit auf der Festplatte rekonstruierbar (Carrier, 2005).

2.3.4 Kopierbarkeit

Digitale Rechensysteme kodieren Daten im binären Zahlensystem. Da Computer alle Informationen schlussendlich im Binärformat speichern, kennen sie nur ***eindeutig unterscheidbare Zustände***. In Anlehnung an die ursprüngliche Bedeutung des Wortes ***diskret*** als ***abgesondert*** verwendet man in der Informatik auch gerne den Begriff der ***diskreten*** Zustände. Diese Bezeichnung betont, dass es zwischen zwei binären Werten keine Zwischenwerte gibt. Folglich befindet sich ein Computer zu jedem Zeitpunkt in einem klar definierten Zustand. Dies steht im Gegensatz zu einer grundlegenden Erfahrung, die man in der realen Welt macht: Materie ist (nahezu) beliebig zerteilbar. Der Zustand der realen Welt ist also alles andere als ***diskret*** im Sinne der Informatik, während der Zustand eines Computers durch die zu jedem Zeitpunkt immer bis ins letzte Bit exakt definiert ist. Im Prinzip kann man alles, also auch alle Arten von Naturphänomenen, wie beispielsweise Bilder oder Geräusche in einer vorher festgelegten Genauigkeit als binäre Zahl kodieren. Dies gilt umso mehr für schriftlich niedergelegte Informationen, Konzepte und Ideen. Die diskrete Form all dieser Artefakte macht es möglich, ***perfekte Kopien*** zu erzeugen, Kopien also, die von ihrem digitalen Inhalt nicht vom Original zu unterscheiden sind.

Wie bereits oben auf Seite → diskutiert, basiert diese Eigenschaft auf der Reduktion der gespeicherten Daten auf die digitale Repräsentation. Aufgrund ihrer digitalen Natur kann man also auch digitale Spureninformationen im Gegensatz zu physikalischen Spuren exakt duplizieren und somit alle Untersuchungen anhand einer Kopie durchführen. Genau genommen kopiert man aber lediglich die digitale Spureninformation, der Spurenträger verändert sich. Dies steht im gefühlten Gegensatz zu vielen physischen Spuren wie etwa Blut oder DNA-Spuren, die für eine Untersuchung chemisch analysiert und darum teilweise zerstört werden müssen.

Wenn man digitale Spuren kopiert, muss man jedoch sicherstellen, dass im Rahmen des Kopiervorgangs die Integrität gewahrt bleibt, also keine Veränderungen an den Spuren stattfinden. Die Übereinstimmung des Originals mit der Kopie lässt sich aber vergleichsweise einfach nachweisen, etwa durch einen bitweisen Vergleich mit dem Original. Wie oben auf Seite → angemerkt, verwendet man in der Praxis oft ***kryptographische Hashfunktionen***, die einen kompakten Fingerabdruck eines Datenträgers erstellen können (für einen Überblick siehe Menezes u. a. (1997) bzw. Kapitel 6).

2.3.5 Semantik

Auch wenn eine formale Unterscheidung schwierig ist, ist es aus Praxissicht sinnvoll, digitale Spuren noch in Bezug auf ihre Semantik auf Anwendungsebene zu klassifizieren. Wir unterscheiden hierzu allgemein die folgenden fünf Arten digitaler Spuren in Form von digitalen Daten.

- Primärdaten
- Sekundärdaten
- Programmdaten
- Konfigurationsdaten
- Logdaten

Die Daten dieser Klassen unterscheiden sich, wie wir im Folgenden erläutern werden, typischerweise in ihrem Inhalt, Entstehungszeitpunkt, Speicherort und zum Teil auch in ihrer Aussagekraft im Kontext einer digital-forensischen Untersuchung. Das bedeutet, dass je nach Art der in einem konkreten Fall gesuchten Spuren eine oder mehrere bestimmte Klassen von Daten zu

untersuchen sind. Diese Unterscheidung ist orthogonal zu den anderen Klassifikationskriterien und kann in bestimmten Anwendungsgebieten verfeinert werden (siehe etwa die fünf Datenkategorien von Festplattendaten von Carrier (2005)).

Primärdaten

Der Begriff ***Primärdaten*** enstammt ursprünglich der Statistik und bezeichnet dort jene Daten, die unmittelbar aus einer Datenerhebung gewonnen werden. Im Falle der forensischen Analyse bezeichnet der Begriff Primärdaten solche Daten, zu deren Verarbeitung eine Anwendung implementiert wurde. Primärdaten sind also in Bezug auf eine konkrete Anwendung jene Daten, die von dieser Anwendung primär verarbeitet werden. Vereinfacht ausgedrückt dient beispielsweise eine konkrete Bildverarbeitungssoftware dazu, Bilddateien zu verarbeiten. Bilddateien sind in diesem Fall Primärdaten. Ein weiteres Beispiel sind E-Mails als Primärdaten eines E-Mail-Clients wie Mozilla Thunderbird[1].

Sekundärdaten

Sekundärdaten werden dazu erstellt und genutzt, um die Verarbeitung von Primärdaten zu vereinfachen. Hierbei kann es sich entweder um Daten handeln, die die Verarbeitung der Primärdaten durch die Anwendung unterstützen (***System-Sekundärdaten***) oder um Sekundärdaten, die dem Benutzer die Bearbeitung der Primärdaten erleichtern (***Benutzer-Sekundärdaten***).

Übliche Beispiele für System-Sekundärdaten sind Caches, Indizes oder Journale, welche die Performanz einer Anwendung bei der Verarbeitung der Primärdaten erhöhen. Systemsekundärdaten werden daher sehr häufig als Nebeneffekt der Verarbeitung von Primärdaten erstellt – meist ohne Absicht oder Kenntnis des Benutzers.

Im Gegensatz dazu werden Benutzer-Sekundärdaten verwaltet, um den Benutzer bei der Arbeit mit Primärdaten durch eine Anwendung zu unterstützen. Diese Daten werden von der Anwendung zur Gewährleistung ihrer Kernfunktionalität, also für die Bearbeitung der Primärdaten, an sich nicht benötigt. Beispiele für Benutzer-Sekundärdaten sind Lesezeichen in Webbrowsern, Adressbücher in E-Mail-Clients oder Listen zuletzt benutzter Dokumente.

Programmdaten

Programmdaten sind jene Daten, welche eine Anwendung selbst ausmachen und werden häufig auch als ***Programmcode*** bezeichnet. Hierbei kann es sich sowohl um eine ausführbare Binärdatei oder Bibliothek, als auch um Bytecode oder auch Quelltext (im Falle einer interpretierten Programmiersprache) handeln. Programmdaten verändern sich meist während der Ausführung der Anwendung nicht – mit einigen Ausnahmen wie beispielsweise einer Updatefunktionalität.

Konfigurationsdaten

Konfigurationsdaten wiederum bestimmen die Art und Weise, in der eine Anwendung Primärdaten verarbeitet. Eine charakteristische Eigenschaft von Konfigurationsdaten ist, dass diese Daten sich meist unmittelbar nach der Installation einer Anwendung oder auf explizite Anforderung des Benutzers verändern und sonst über die meiste Zeit konstant bleiben.

Es gibt außerdem Konfigurationsdaten, auf die nicht direkt Einfluss genommen werden kann. Beispielsweise werden bei der Installation einer Anwendung Verzeichnisstrukturen angelegt. Hierzu zählen beispielsweise Ordner, in denen bearbeitete Primärdaten standardmäßig gespeichert werden. Zudem gibt es für jedes Betriebssystem spezifische Verzeichnisse, in welchen die Programmdaten oder auch die Konfigurationsdaten selbst gespeichert werden. Solche Konfigurationsdaten bezeichnen wir als ***implizite Konfigurationsdaten***.

Im Gegensatz zu den Konfigurationsdaten, die von der Anwendung festgelegt oder durch den Benutzer angepasst werden, gibt das verwendete Betriebssystem auch implizite Angaben vor. Hierzu zählen:

- Speicherort von temporären Daten
- Verzeichnisnamen, zum Beispiel Name des Standardordners für Programmdaten oder Benutzerverzeichnisse
- Betriebssystemabhängige Verwaltungsstrukturen

Diejenigen Konfigurationsdaten, die explizit vom Benutzer verändert werden können, um das Verhalten der Anwendung zu steuern, bezeichnen wir im

Gegensatz zu den impliziten Konfigurationsdaten als ***explizite Konfigurationsdaten***.

Logdaten

Viele Anwendungen sichern ein Protokoll von ausgeführten Aktionen, Fehlermeldungen oder Informationen zur Fehlerbehebung als ***Logdaten***. Diese Daten ermöglichen es beispielsweise einem Administrator das Verhalten der Anwendung nachzuvollziehen. Jedoch sind Logdaten für die Anwendung selbst in keiner Weise von Belang und entstehen nur, wenn die entsprechende Log-Funktionalität willentlich implementiert wurde. Auch für einen gewöhnlichen Benutzer, der ausschließlich an der Verarbeitung der Primärdaten interessiert ist, bieten diese Daten keinen Mehrwert. Falls solche Daten vorhanden sind, bieten sie jedoch wertvolle Rückschlüsse über vergangene Aktivitäten des Systems und sind daher im Kontext einer forensischen Untersuchung von besonderer Relevanz.

2.4 Assoziation mittels digitaler Spuren

In den klassischen forensischen Wissenschaften (siehe Kapitel 1) bezeichnet der Begriff ***Assoziation*** den Vorgang, bei dem zwei Objekte in Beziehung zueinander gesetzt werden. In der physischen Welt war dies zumeist ein „Kontakt“ zwischen zwei Objekten, durch den man eine Zuordnung machen konnte (etwa Projektil zu Waffe, Schuhabdruck zu Schuh). Als Ergebnis der Assoziation steht ein ***Ereignis***, das ein Puzzleteil zur Rekonstruktion des Tathergangs beiträgt.

Wir haben oben gesehen, dass viele Aspekte digitaler Spuren ein Analogon zu Aspekten physischer Spuren haben. Im Folgenden zeigen wir, dass auch der Prozess der Assoziation analog zu jenem aus der klassischen Forensik durchschritten werden kann. Ziel ist die Assoziation zweier Datenobjekte (Bitfolgen) auf Basis von Musterübertragung.

2.4.1 Identifizierung

Bei der ***Identifizierung*** wird die prinzipielle Tauglichkeit der Spur als Beweismittel geprüft. Die Spur selbst kann eine beliebige Bitfolge sein, die

potentiellen Tatbezug aufweist, etwa weil sie auf einer Festplatte liegt, die am Tatort gefunden wurde. Aber nicht alle Bitfolgen müssen auch Spuren sein. Die Identifizierung unterscheidet diese beiden Fälle. Ziel der weiteren Schritte ist nun, das „Gegenstück“ der gefundenen Spur zu finden, also beispielsweise die Quelle, aus der die Bitfolge stammt, oder die Umstände/Aktionen, die beim Entstehen der Bitfolge stattgefunden haben müssen.

2.4.2 Klassifizierung

Bei der ***Klassifizierung*** einer Spur wird die Menge der Gegenstücke weiter eingegrenzt durch eine genauere Analyse klassifizierender Merkmale. Charakteristisch für die Klassifizierung ist die Tatsache, dass man meist spezielle Werkzeuge benötigt, um solche Merkmale auszumachen. Klassifizierungsmerkmale entstehen aus ***kontrollierten*** Herstellungsprozessen, die dann jeweils charakteristisch für die Klasse von Objekten sind. Im Falle digitaler Spuren sind etwa Dateinamenssuffixe (die den Typ einer Datei andeuten) klassifizierende Merkmale. Oder die Entropie einer Datei kann ein klassifizierendes Merkmal sein.

2.4.3 Individualisierung

Mit ***Individualisierung*** bezeichnet man die Zuordnung der Spur zu einer eng umgrenzten Menge von Gegenstücken, die potentiell in Verbindung zur Spur stehen. Idealerweise führt bereits die Individualisierung zu einer 1:1-Zuordnung zwischen Spur und dem Gegenstück. Die Merkmale, die eine Individualisierung erlauben (***individualisierende Merkmale***), entstammen definitionsgemäß ***zufälligen*** und ***unkontrollierten*** Prozessen.

Individualisierende Merkmale im Bereich der digitalen Spuren sind beispielsweise Nutzungsspuren, die durch menschliche Benutzer verursacht worden sind, also Inhalte von Dokumenten oder Logdateien menschlicher Interaktionen. Auch Spuren von Systemaktivitäten könn individualisierend sein, beispielsweise die Namen von temporären Dateien oder die Wahl zufälliger Schlüssel bei der Datenübertragung.

2.4.4 Assoziation

Auf Basis der vorherigen Schritte steht im Optimalfall am Ende dieses Prozesses eine ***Assoziation*** zwischen dem am „Tatort“ identifizierten Objekt ***A*** und einem anderen Objekt ***B***, welches mit der Spur in Zusammenhang steht.

In der digitalen Welt kann man nur den Kontakt von ***digitalen*** Objekten feststellen. Formal werden hierbei Ähnlichkeiten in den Speicherinhalten zugrunde gelegt, die den Zustandsraum des Computers ausmachen. Findet man beispielsweise eine inhaltsgleiche Datei an zwei Orten (im gleichen Dateisystem oder auf Festplatten unterschiedlicher Rechner), dann kann man die Hypothese einer ***gemeinsamen Quelle*** aufstellen.

Um die Art der Assoziation in der digitalen Welt besser zu veranschaufichen, betrachten wir nun anhand verschiedener Beispiele diesen Prozess. Hierbei möchten wir insbesondere zeigen, wie sich die Theorie von Inman u. Rudin (2000) auf digitale Spuren anwenden lässt und zum Teil implizit bereits angewandt wird.

Wir beginnen bewusst mit zwei sehr einfachen Beispielen, bei denen es um den Kontakt zwischen physischen Objekten geht. Daher ist der Kontakt in diesen Beispielen offensichtlich und die Assoziation intuitiv nachvollziehbar, auch wenn der eigentliche Vorgang der Assoziation in der digitalen Welt stattfindet. Daraufhin folgt in Abschnitt 2.4.7 ein etwas weniger offensichtlicher Fall einer Assoziation, die ausschließlich zwischen digitalen Objekten besteht. Am Ende bringen wir ein akademisches Beispiel, das den Vorgang auf den minimalen Kern reduziert.

2.4.5 Beispiel: Multimediaforensik

Als erstes Beispiel für eine Assoziation in der digitalen Welt möchten wir ein Beispiel aus dem Bereich der ***Multimediaforensik*** heranziehen. In der Multimediaforensik werden solche digitalen Spuren betrachtet, die durch einen Sensor aus der physischen Welt aufgezeichnet werden, wie beispielsweise Digitalfotografien. Artefakte, die hierbei vom Sensor in diesen Daten hinterlassen werden, können genutzt werden, um Fragen nach Konsistenz und Ursprung der Daten zu beantworten. Die Feststellung der Herkunft, also beispielsweise Aussagen wie „Bilddatei ***A*** wurde mit Digitalkamera ***B*** aufgenommen“, ist ein offensichtliches Beispiel für die Herstellung einer Assoziation, in diesem Fall zwischen einem digitalen Medienobjekt und dem Sensor, der es ursprünglich aufgezeichnet hat. Anhand dieses Beispiels vollziehen wir nun die Schritte der ***Identifikation, Klassifikation*** und

Individualisierung hin zur ***Assoziation***. Mehr Hintergrund zum Bereich Multimediaforensik gibt Kapitel 4.

Identifikation

In diesem Beispiel fassen wir eine Datei auf einer sichergestellten Festplate oder einem bestimmten Server im Internet als „am Tatort“ aufgefundenes Objekt ***A*** auf. Diese Datei wird als potentielle Spur identifiziert. Gesetzt den Fall, es handelt sich bei der betrachteten Ermittlung um einen Fall des Besitzes illegaler pornografischer Schriften, so ist bereits allein die Existenz dieser Datei möglicherweise ausreichend, um Objekt ***A*** als potentielles Beweismittel einzustufen.

Klassifizierung

Anschließend wird das identifizierte Objekt ***A*** (die Datei) auf klassifizierende Charakterisika untersucht. In diesem konkreten Beispiel würde dies zu dem Resultat führen, dass Objekt ***A*** eine Bilddatei einer bestimmten Größe (Auflösung) in einem bestimmten Dateiformat (beispielsweise JPEG) ist. Eine kurze Analyse der Datei (beispielsweise durch Auswertung der EXIF-Daten) führt zu dem Schluss, dass das Bild wahrscheinlich durch eine Digitalkamera (das zu findende Objekt ***B***) aufgezeichnet wurde. Durch Auswertung weiterer Klassencharakteristika, wie beispielsweise kleinen aber systematischen Abweichungen der Lichtempfindlichkeit einzelner Sensorelemente, ist es häufig möglich, die Klasse anhand einer einzelnen Fotografie auf den Hersteller und unter Umständen sogar auf das Modell der Kamera einzuschränken (Chen u. a., 2008).

Individualisierung

Wurden mehrere Digitalkameras im Haus eines Verdächtigen sichergestellt, so ist es im nächsten Schritt möglich, das exakte Rauschverhalten einer jeden Kamera zu untersuchen. Dies geschieht durch die Aufnahme einer Reihe von Bildern unter kontrollierten Bedingungen in einem Labor, aus denen sich das individuelle Rauschmuster der jeweiligen Kamera extrahieren lässt (Chen u. a., 2008; Lukás u. a., 2005). Durch Überprüfung von Objekt ***A*** auf die

charakteristischen Muster dieser Kameras, kann dann die Menge der für die Aufzeichnung von Objekt ***A*** in Frage kommenden Digitalkameras im Optimalfall auf ein einzelnes Gerät eingegrenzt werden.

Assoziation

Nach den vorhergehenden Schritten ist es nun durch die individuellen (genauer: individualisierenden) Merkmale jeder Kamera möglich, mit hoher Sicherheit eine Assoziation zwischen dem identifizierten Bild (Objekt ***A***) und einer bestimmten Digitalkamera herzustellen.

2.4.6 Beispiel: USB-Speichergeräte

Als nächstes Beispiel betrachten wir die Assoziation zwischen einem Wechseldatenträger und einem bestimmten Computer. Speichergeräte werden heute meist über den ***Universal Serial Bus*** (USB) an einen Computer angeschlossen. Bekanntermaßen sammeln Betriebssysteme Informationen über an das System angeschlossene Datenträger. Tatsächlich nutzen Betriebssysteme die individuellen Charakteristika eines Speichergerätes, um beispielsweise den korrekten Gerätetreiber auszuwählen (Carvey u. Altheide, 2005). Diese Tatsache kann es ermöglichen, Aussagen wie „USB-Stick ***A*** war schon einmal mit Computer ***B*** verbunden“ zu treffen.

Identifikation

Objekt ***A*** ist hier ein kleines Plastikobjekt mit einem Metallende, welches am Tatort aufgefunden wird. Wir nehmen für dieses Beispiel an, dass es sich um eine Ermittlung wegen Datendiebstahl handelt. Dies sollte ausreichen, um Objekt ***A*** als potentielles Beweismittel zu identifizieren.

Klassifizierung

Eine weitergehende Untersuchung auf klassifizierende Merkmale des Gerätes ergibt, dass es sich bei Objekt ***A*** tatsächlich um ein USB-Massenspeichergerät einer bestimmten Marke handelt. Die Klasse der Objekte ***B***, zu denen Objekt ***A***

assoziiert werden könnte, umfasst zu diesem Zeitpunkt die Menge aller Computersysteme, die über eine USB-Schnittstelle verfügen.

Individualisierung

Stellen wir uns eine Menge von Computern verschiedener Verdächtiger vor, welche ebenfalls sichergestellt wurden, und stellen wir uns vor, dass auf all diesen Geräten Microsoft Windows als Betriebssystem eingesetzt wird. Sobald ein USB-Speichergerät an ein Windows-System angeschlossen wird, erstellt das Betriebssystem eine sogenannte ***Geräteinstanzkennung*** (***device instance identifier***) auf Basis unterschiedlicher auf dem Gerät hinterlegter Werte, wie beispielsweise einer eindeutigen Seriennummer. Derartige Informationen stehen auf nahezu allen Geräten zur Verfügung. Diese Geräteinstanzkennungen werden unter Windows in der ***Windows Registrierung*** (***Registry***) unter folgendem Schlüssel gespeichert:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\USBStor
```

Diese Daten können unter Einsatz entsprechender Software extrahiert werden (Carvey u. Altheide, 2005). Weitere individualisierende Merkmale könnten in einer bestimmten Menge von Dateien bestehen, die sich sowohl auf Objekt ***A***, als auch auf Objekt ***B*** befinden.

Assoziation

Das Auffinden der einzigartigen Gerätekennung eines USB-Speichergerätes an einer bestimmten Stelle auf einem Computersystem ist ein starkes Indiz dafür, dass dieses konkrete Gerät (Objekt ***A***) in der Vergangenheit einmal an diesen konkreten Computer (Objekt ***B***) angeschlossen war.

2.4.7 Beispiel: Browser Cache

Aus Performanz-Gründen sichern Webbrowser meist lokale Kopien besuchter Webseiten in Cache-Dateien, um diese bei einem erneuten Aufruf der Webseiten nicht nochmals herunterladen zu müssen. Diese Dateien gehören zu den System-Sekundärdaten (siehe Abschnitt 2.3.5) und können von forensischen Wissenschaftlern genutzt werden, um zu rekonstruieren, welche Webseiten von einem Benutzer eines Computers in der Vergangenheit aufgerufen wurden. Das

bedeutet, dass Cache-Dateien es ermöglichen, eine Verbindung zwischen einem bestimmten Computer (Objekt ***A***) und einer konkreten Webseite (Objekt ***B***) herzustellen. Der Prozess der Assoziation kann in einem solchen Fall (und wird in den meisten Fällen bereits implizit) auf folgende Art und Weise angewandt werden.

Identifikation

Auf den ersten Blick könnte der forensische Wissenschaftler Dateien unter einem bestimmten Pfad im Dateisystem vorfinden, von dem bekannt ist, dass er üblicherweise von einem bestimmten Webbrowser für die Sicherung von Cache-Dateien genutzt wird. Für den ***Microsoft Internet Explorer*** ist dies beispielsweise das folgende Verzeichnis:

```
%systemdir%\Documents and Settings\%Username%\Local
   Settings\Temporary Internet Files\Content.ie5
```

Daher könnte der Forensiker in diesem Schritt zu dem Schluss kommen, dass diese Dateien potentiell zum Browser-Cache gehören könnten und sie daher als potentielle Spuren identifizieren.

Klassifizierung

Im nächsten Schritt benötigt der Forensiker typischerweise irgendeine Art von (Software-) Werkzeug, um weitere Schlüsse ziehen zu können. Abhängig von dem Webbrowser, der die Cache-Dateien erzeugt hat, kann beispielsweise ein spezieller Parser für das konkrete Dateiformat erforderlich sein, um zu verifizieren, dass die identifizierten Dateien tatsächlich zum Cache des betreffenden Browsers gehören.

Diese Spuren entstehen durch einen bekannten und kontrollierten Herstellungsprozess, da dieser Webbrowser bekanntermaßen immer (so lange nicht manuell anderweitig konfiguriert) Cache-Dateien unter diesem bestimmten Pfad und Dateinamen in diesem konkreten Format speichert. Dies ist ein wichtiges Kriterium für die Unterscheidung der Klassifizierung und der Individualisierung. Das Ergebnis dieses Schrittes ist, dass diese Dateien tatsächlich Cache-Dateien dieses Webbrowsers sind.

Individualisierung

Schließlich werden die eigentlichen Inhalte der als Cache klassifizierten Dateien untersucht. Hierzu kommen meist wiederum Werkzeuge zum Einsatz, wie in diesem Fall möglicherweise eine Bildbetrachtungssoftware zur Darstellung gecachter Bilddateien oder ein Webbrowser, um zwischengespeicherte HTML-Dateien zu rendern. In diesem Schritt versucht der Forensiker herauszufinden, um welche konkreten Inhalte es sich handelt. In Abhängigkeit vom bereits erwähnten Dateiformat der Dateien und dem eingesetzten Betriebssystem, können auch weitere Informationen, wie beispielsweise Zeitstempel oder der Name des Benutzers, der das Cachen der Webseite verursacht hat, zur Verfügung stehen. Auch die Herkunft der Inhalte, also die URL von der sie durch den Browser heruntergeladen wurden, können ausgelesen werden.

Diese Art von Spuren unterliegt für gewöhnlich einem unbewussten Entstehungsprozess, da die Tatsache, dass ein Benutzer mit diesem konkreten Benutzernamen das Cachen dieser konkreten Webseite mit diesen Inhalten zu genau dieser Zeit anstößt, höchst individuell ist. Daher ist es als unwahrscheinlich anzusehen, dass sich ein zweiter Computer (insbesondere im Kreis der Verdächtigen) findet, der exakt die gleichen (und nur diese) Spuren aufweist. Als Ergebnis der Individualisierung würde der forensische Wissenschaftler feststellen, dass auf dem untersuchten System dieser konkrete Webbrowser zur spezifizierten Zeit diese Inhalte geladen und daher gecacht hat.

Bei Browsern neuester Generation könnte diese Aussage zum Teil nur weniger scharf formuliert werden, da zum Teil Inhalte von Webseiten im Voraus geladen werden, wenn sie von der aktuell betrachteten Seite verlinkt werden.

Assoziation

Als Ergebnis der vorausgegangenen Schritte ist der Forensiker in der Lage, eine Assoziation zwischen einem konkreten Benutzer des untersuchten Systems und einer Webseite herzustellen. Außerdem kennt er den Zeitpunkt des Ereignisses (den Aufruf der Website) und die geladenen Inhalte.

2.4.8 Beispiel: Copy/ Move Operationen im Hauptspeicher

Die vorhergehenden Beispiele berührten immer in irgendeiner (wenn auch schwachen) Form einen „physischen Kontakt“, sei es unmittelbar sichtbar wie im ersten Beispiel oder durch eine Netzverbindung wie im dritten Beispiel. In diesem Beispiel möchten wir zeigen, dass sich auch der reine Transfer von

Mustern in Form einer Kopieroperation eines Computers (also auf niedrigster Ebene einer ***copy***- oder ***move***-Maschineninstruktion) in den Prozess der Assoziation fügt. Allerdings müssen hierzu zunächst einige Annahmen über die Systemumgebung getroffen beziehungsweise definiert werden, um die zweifelsfreie Herstellung einer Assoziation zu ermöglichen. Daher ist das letzte Beispiel eher künstlicher Natur und weniger intuitiv verständlich als die anderen Beispiele. Auch scheint es auf den ersten Blick mit der Praxis wenig zu tun zu haben. Jedoch ist es mit Bedacht auf den minimalen Kern reduziert und zeigt die Assoziation eines Speicherbereiches (Objekt ***A***) mit einem anderen Speicherbereich (Objekt ***B***) eines Computers als integralen Bestandteil aller zuvor aufgeführten Beispiele.

Im Gegensatz zu realen Computern erlaubt der hier betrachtete, abstrakte Automat drei anstatt nur zwei unterschiedliche Werte für jede Speicherstelle: Neben den gebräuchlichen binären Werten 0 und 1, führen wir einen expliziten NIL-Wert $\perp$ ein. Unser Automat ist mit einem 2-Bit read-only Speicher (ROM) ausgestattet: Das erste Bit, welches wir mit ***A*** bezeichnen, speichert den Wert 0, während die zweite Speicherstelle ***B*** den Wert 1 enthält. Weiterhin operiert der Rechner auf einem 1-Bit randomaccess Speicher (RAM) ***R***, der mit dem Wert $\perp$ initialisiert ist. Wir nennen die aktuelle Zuweisung von Werten zu diesen drei Speicherstellen den ***Zustand*** des Automaten.

Der einzige Befehl, der auf dieser Maschine ausgeführt werden kann, ist der ***move***-Befehl `mov`, der den Wert einer Ursprungs-Speicherstelle in eine Ziel-Speicherstelle kopiert. Die Syntax dieses Kommandos lautet wie folgt: `mov (Quelle), (Ziel)`. Als Ursprungs-Speicherstelle kann jede der 3 Speicherstellen des Automaten (***A, B*** und ***R***) genutzt werden. Als Ziel der Operation kann aber ausschließlich der RAM ***R*** zum Einsatz kommen, da der ROM-Speicher feste Werte enthält und nicht beschrieben werden kann. Schließlich nehmen wir an, dass es keinerlei externe Einflüsse wie beispielsweise Strahlung gibt, welche Einfluss auf den Automaten nehmen und ohne Ausführung eines Befehls dessen Zustand verändern könnten. Abbildung 2.2 veranschaufichet zum einen den Initialzustand des Automaten in der linken Hälfte der Abbildung und zum anderen den Zustand, nachdem die Operation `mov` ***B, R*** ausgeführt wurde (rechter Teil der Abbildung).

Als Beispiel für die Assoziation betrachten wir nun den Fall, in dem wir den Automaten im Zustand nach der Ausführung besagten Kommandos vorfinden, wie er in Abbildung 2.2 dargestellt wird.

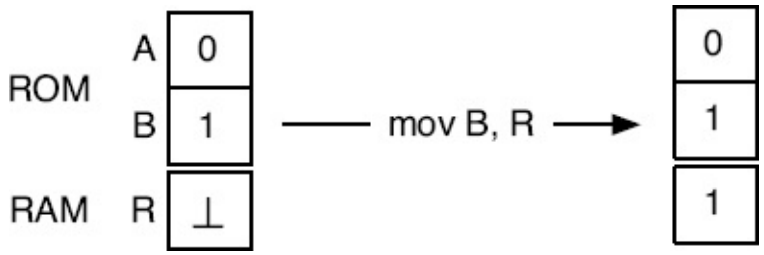


Abbildung 2.2: Zustand des 3-Bit Automaten vor und nach der Ausführung eines `mov` Befehls.

Identifikation

Wenn wir den gegebenen Zustand untersuchen, können wir zunächst feststellen, dass der RAM im Gegensatz zum bekannten initialen Zustand des Systems nicht den Wert $\perp$ enthält. Daher ziehen wir die Schlussfolgerung, dass RAM $\boldsymbol{R}$ Spuren eines Ereignisses innerhalb des Systems aufweist und identifizieren diesen als potentielles Beweismittel für dieses Ereignis.

Klassifizierung

Im nächsten Schritt könnten wir die identifizierten Spuren als Bit $\boldsymbol{R} \in \{0,1\}$ klassifizieren und schlussfolgern, dass es eine Kopie einer Speicherstelle ist, welche nicht den Wert $\perp$ enthält. Obgleich möglicherweise nicht offensichtlich, so ist dies eine klassifizierende Eigenschaft, da bekannt ist, dass $\boldsymbol{R}$ in unserem System immer ein Bit enthält, nachdem ein `mov`-Kommando ausgeführt wurde. Weiterhin ist diese Eigenschaft noch nicht individualisierend, da wir weder das exakte Kommando (mit Quell- und Zieloperator) kennen, noch die Speicherstelle bekannt ist, aus der dieser konkrete Wert stammt.

Individualisierung

Im Schritt der Individualisierung untersuchen wir wiederum den eigentlichen Inhalt des RAMs und stellen fest, dass $\boldsymbol{R}$ den konkreten Wert 1 enthält. Daher wissen wir nun, dass diese Spur ausschließlich eine Kopie einer anderen Speicherstelle, die diesen Wert enthält, sein kann.

Assoziation

Aufgrund des Vorwissens über unseren 3-Bit Automaten können wir schließen, dass das in ***R*** gespeicherte Bit ausschließlich von ROM ***B*** stammen kann, da dies die einzige Speicherstelle im System ist, die neben ***R*** den Wert 1 enthält. Daher können wir nun schlussfolgern, dass die Operation `mov` ***B, R*** ausgeführt wurde (was unser rekonstruiertes Ereignis ist) und diese Spur hinterlassen hat. Wir können weiterhin die Irrtumswahrscheinlichkeit für diese Aussage spezifizieren: Die Wahrscheinlichkeit, dass `mov` ***B, R*** nicht ausgeführt wurde, ist in diesem Modell 0, da es keinerlei äußere Einflüsse gibt und die einzige Möglichkeit, den Wert des RAMs zu verändern die Ausführung der `mov` Operation ist.

Es kann selbstverständlich sein, dass außerdem weitere Operationen stattgefunden haben, wie beispielsweise zuvor `mov` ***A, R***. Es könnte sogar im Anschluss an das rekonstruierte Ereignis die Aktion `mov` ***R, R*** ausgeführt worden sein. Die Aussage, dass die Operation `mov` ***B,R*** ausgeführt wurde, wird hiervon jedoch nicht beeinträchtigt. Dies liegt an den stark individualisierenden Merkmalen der Aktion. Wir werden in Kapitel 3 diesen Begriff unter dem Terminus der charakteristischen Spur formalisieren.

2.4.9 Quantifizierung der Irrtumswahrscheinlichkeit

Ein wesentlicher Bestandteil der Assoziation ist die Quantifizierung der Irrtumswahrscheinlichkeit. Die Irrtumswahrscheinlichkeit beziffert die „Überzeugungskraft“ der Assoziation im Gerichtsverfahren. Die Irrtumswahrscheinlichkeit beziffert die Wahrscheinlichkeit, dass kein Kontakt stattfand, obwohl dieser festgestellt wurde. Die Berechnung der Irrtumswahrscheinlichkeit ist wie auch im Bereich der klassischen Forensik immer abhängig vom Kontext. Meist müssen sehr viele Annahmen getroffen werden, um zu einer Berechnungsgrundlage zu kommen.

Statt quantitative Angaben machen zu wollen, definiert Casey (2011, S. 69f) sieben ***qualitative*** Grade von Wahrscheinlichkeit (***levels of certainty***). Diese Stufen verbindet er mit sprachlichen Ausdrücken, die in Gutachten oder vor Gericht verwendet werden können. Wir betrachten diese sieben Grade nun im Einzelnen:

- Das Ereignis ist ***fehlerhaft / inkorrekt*** (Grad C0).
  Die Spuren widersprechen bekannten Fakten oder stimmen nicht überein. Beispielsweise werden bestimmte Aktivitäten wie der Versand von Spam durch bekannte Formen von Schadsoftware durchgeführt. Die Behauptung,

eine Person hätte absichtlich und manuell diese Spam-Nachrichten verschickt, wäre fehlerhaft.

- Das Ereignis ist ***sehr unwahrscheinlich*** (C1).
  Die Spuren sind fragwürdig.
- Das Ereignis ist ***unwahrscheinlich*** (C2).
  Es gibt nur eine Quelle für die digitale Spur und diese Quelle war nicht geschützt vor Manipulationen.
- Das Ereignis ist ***möglich*** (C3).
  Die Quelle(n) der digitalen Spur sind schwerer zu manipulieren als im Fall C2 aber es gibt entweder nicht genügend Spuren oder die Spuren sind inkonsistent.
- Das Ereignis ist ***wahrscheinlich*** (C4).
  Entweder ist die digitale Spur gegen Manipulationen geschützt gewesen, oder es existieren mehrere unabhängige und übereinstimmende digitale Spuren. Beispiele sind konsistente Spuren auf der Festplatte eines mutmaßlichen Erpressers: die Erpressungs-E-Mail als Kopie im Postausgang sowie mehrere Entwürfe derselben E-Mail und elektronische Kontoauszüge, die den Eingang des erpressten Geldes auf dem Konto des Beschuldigten dokumentieren.
- Das Ereignis ist ***sehr wahrscheinlich*** (C5).
  Es gibt mehrere unabhängige und übereinstimmende digitale Spuren, die zudem vor Manipulationen geschützt waren. Allerdings existieren kleinere Inkonsistenzen, beispielsweise leichte Abweichungen der Zeitstempel. Wenn etwa die IP-Quelladresse von bösartigem Netzwerkverkehr auf einen Anschluss ***X*** aufgelöst wird und zusätzlich durch die Überwachung von Netzwerkverkehr festgestellt wird, dass derselbe Verkehr vom Anschluss ***X*** kommt, dann ist es sehr wahrscheinlich, dass die Aktivitäten in der Tat über den Anschluss ***X*** ins Netz gelangten.
- Das Ereignis ist ***sicher*** (C6).
  Die digitalen Spuren waren vor Manipulationen geschützt oder haben hohe statistische Konfidenz. Beispiele sind Funde von inkriminierenden Dateien auf einer Festplatte, die neben einer visuellen Inspektion auch über ihre kryptographischen Fingerabdrücke (mittels kryptographischer Hashfunktionen) als solche identifiziert werden können.

Tabelle 2.1 stellt die deutschen und englischen Begriffe der verschiedenen Grade gegenüber. Für die Praxis erscheinen vor allem die Stufen C0 sowie C4 bis C6 relevant. Hier muss man allerdings differenziert argumentieren.

Grad	Ausdruck (Deutsch)	Ausdruck (Englisch)
C0	inkorrekt	***erroneous/incorrect***
C1	sehr unwahrscheinlich	***highly uncertain***
C2	unwahrscheinlich	***somewhat uncertain***
C3	möglich	***possible***
C4	wahrscheinlich	***probable***
C5	sehr wahrscheinlich	***almost certain***
C6	sicher	***certain***

Tabelle 2.1: Deutsche und englische Begriffe für Qualitative Grade von Wahrscheinlichkeit nach Casey (2011).

Casey (2011) betont, dass diese Stufen weiterhin subjektiv bleiben: Verschiedene Ermittler können auf Basis derselben Spuren zu unterschiedlichen Einschätzungen kommen. Insofern sei die Skala auch nur als ein erster Ansatz zu verstehen, welcher noch weiter erforscht werden muss. Generell basiert aber bereits die Quantifizierung einer relativ einfache Aussage wie „Der Nutzer hat Datei ***x*** über die Tauschbörse ***y*** bezogen“ auf einer Vielzahl von Annahmen. Sie alle aufzuzählen und darauf eine komplexe Berechnung aufzubauen, erscheint generell möglich, ist aber nur für sehr eingeschränkte und sehr exakt fassbare Aussagen sinnvoll (Overill u. a., 2013; Overill u. Silomon, 2012; Overill u.a., 2010). Zudem muss wie in der klassischen Forensik zwischen zwei Wahrscheinlichkeiten unterschieden werden:

1. Die Irrtumswahrscheinlichkeit bei der Interpretation einer digitalen Spur, und
2. die Wahrscheinlichkeit, mit der die digitale Spur unverfälscht vorliegt.

In der klassischen Forensik gibt es im Kontext der vermeintlich so objektiven DNA-Analyse immer wieder Pannen, die vor allem die zweite Wahrscheinlichkeit beeinflussen (etwa die Kontamination des Spurenträgers wie beim „Phantom von Heilbronn“ (Schule, 2008)). Gerade die zweite

Wahrscheinlichkeit muss bei der Anfälligkeit digitaler Spuren gegenüber Manipulationen besonders beachtet werden.

Sinnvoll ist die Quantifizierung der Irrtumswahrscheinlichkeit also nur bei sehr einfachen Fragestellungen, die man präzise als Assoziationen ausdrücken kann und für die einfache Basisannahmen getroffen werden können. Komplexere Aussagen werden insbesondere durch die Summe möglichst vieler unabhängiger (und untereinander konsistenter) Spuren überzeugend, auch wenn für sie keine exakte Irrtumswahrscheinlichkeit anzugeben ist. In diesem Sinne kann auch der oben beschriebene Ansatz von Casey interpretiert werden.

2.5 Assoziation als zentraler Aspekt der forensischen Informatik

Nachdem wir gesehen haben, dass das Konzept der Assoziation auch im Bereich der digitalen Spuren sinnvoll ist, setzen wir diese Einsicht in den Kontext allgemeiner Betrachtungen zur Praxis und zur Definition des Gebietes der forensischen Informatik.

2.5.1 Computerforensik

Die ***Computerforensik*** (Geschonneck, 2006) umfasst in ihrer aktuellen Form eine Vielzahl unterschiedlichster Aufgaben, wie beispielsweise

- die möglichst schnelle Bewertung eines Sicherheitsvorfalls anhand erster durch Techniken der ***Live-Analyse*** (Carrier, 2005, S. 13 ff.) erhobener Daten zur Planung der weiteren Untersuchung des Vorfalls.
- die Anfertigung einer forensischen Kopie physischer Speichermedien unter Einsatz spezieller Hard- und Software.
- die Umgehung von Schutzmechanismen digitaler Systeme, um eine Erhebung von Daten zu ermöglichen.
- die Extraktion kryptographischer Schlüssel aus Hauptspeicherabbildern, zur Erhebung verschlüsselter Daten.
- die Rekonstruktion gelöschter Daten anhand von Dateisystem Metadaten oder durch ***Filecarving***.

- die Erstellung von Timelines untersuchter Systeme, also die Erfassung einer zeitlichen Abfolge vergangener Ereignisse auf dem untersuchten System.

Ein weiteres Problemfeld der Computerforensik entspringt der Notwendigkeit mit immer größeren Datenmengen umgehen können zu müssen (Bäcker u. a., 2010). Dies umfasst sowohl die reine Gewinnung, Speicherung und Verarbeitung der Daten, als auch die Aufgabe in der beständig steigenden Datenmenge diejenigen Objekte zu identifizieren, die für die Untersuchung inhaltlich relevant sind.

In den meisten Fällen sind Computerforensiker stark in den Ermittlungsprozess einbezogen oder leiten ihn sogar. Sie erheben erste Spuren, beziehen diese auf den konkreten Fall und ziehen daraus Schlüsse, um weitere Schritte festzulegen. Hierbei spielt sowohl die Erfahrung der Experten, als auch ihr Wissen über funktionale Zusammenhänge und Prinzipien in digitalen Systemen eine wichtige Rolle. Dies gilt insbesondere im Zusammenhang mit ***Incident Response*** (Mandia u. a., 2003) im Unternehmensumfeld, wo die Verantwortung für die gesamte Untersuchung in den Händen von sogenannten CERTs (***Computer Emergency Response Teams***) liegt. Aber auch in der Strafverfolgung werden viele Ermittlungen durch speziell ausgebildete Experten geleitet. ***Alle*** diese Tätigkeiten fallen heute in den Bereich der Computerforensik (siehe Casey (2011); Mandia u.a. (2003); Garfinkel (2010)).

Vergleicht man nun die Computerforensik mit der in Kapitel 1 beschriebenen Theorie klassischer Forensik, so lässt sich ein gewisses Missverhältnis zwischen der Art von Aufgaben, mit denen sich die Forensiker in den beiden Gebieten befassen, feststellen. Während in der Theorie von Inman u. Rudin (2000) die Begriffe ***Transfer*** und ***Assoziation*** den zentralen Aspekt jeder forensischen Fragestellung darstellen, scheint die Computerforensik einen wesentlich weiter gefassten Fokus zu besitzen. Carrier u. Spafford (2004) erklärt dies „historisch“ durch einen „wesentlich komplexeren Prozess, bei dem der Ermittler die Aktivitäten des Benutzers nachverfolgen muss und keine einfache Ja-oder-Nein-Antwort geben kann“ („... ***much more involved process where the investigator must trace user activity and cannot provide a simple yes or no answer“*** (Carrier u. Spafford, 2004)). Lässt sich die zentrale Theorie des Transfers jedoch auf die meisten Fragestellungen der Computerforensik nicht übertragen, so stellt sich die Frage, ob die Computerforensik überhaupt als forensische Wissenschaft (im Sinne von Inman u. Rudin (2000)) bezeichnet werden kann.

Einen ersten Anhaltspunkt für eine beginnende Differenzierung in der Forschungsgemeinschaft der Computerforensik liefern Böhme u.a. (2009), die

feststellen, dass Multimediaforensik nicht gleich Computerforensik ist. Die Multimediaforensik befasst sich mit bestimmten Arten digitaler Spuren, wie digitale Fotos, Videos oder anderen Arten von Aufnahmen. Böhme u. a. (2009) argumentieren, dass diese Multimediadaten mittels Sensoren aus der physischen Welt aufgezeichnet werden und nicht ***vollständig*** in einem digitalen System entstehen, wie es beispielsweise bei E-Mails, Text-Dokumenten oder Log-Dateien der Fall ist. In der Multimediaforensik spielen also individualisierende Merkmale der physikalische Aufzeichnungseinheit eine zentrale Rolle.

2.5.2 Forensische Informatik

Wir definieren allgemein ***forensische Informatik*** als die Anwendung wissenschaftlicher Methoden der Informatik auf Fragen des Rechtssystems. Insbesondere stellt die forensische Informatik Methoden zur gerichtsfesten Sicherung und Verwertung digitaler Spuren bereit. Unsere Definition stellt die Informatik stärker als bisher in den Kontext anderer forensischer Wissenschaften. Die Betonung des wissenschaftlichen Aspekts ist außerordentlich wichtig, da eine forensische Beweisführung dazu führen kann, dass Menschen ihrer Freiheit beraubt werden. Nur eine verlässliche und objektive (wissenschaftliche) Methodik wird dieser Verantwortung gerecht. Die angewendete Methodik muss daher auch immer wieder neu hinterfragt und überdacht werden. Eine besondere Rolle spielt die Unterfütterung des Gebietes mit theoretischen Grundlagen der klassischen Forensik, und ein tieferes Verständnis der Art von Spuren, die in der forensischen Informatik untersucht werden.

Wir unterscheiden zwei Aspekte der forensischen Informatik und teilen die relevanten Methoden in zwei Gebiete:

1. Die forensische Informatik im ***engeren Sinne*** umfasst eine sehr fokussierte Menge von Fragestellungen und Methoden, welche sich im Kern mit der Herstellung von Assoziationen, wie sie zuvor geschildert wurden, befasst. Weiter unten in Abschnitt 2.5.3 geben wir eine Reihe von Beispielen solcher Fragestellungen, um den Fokus dieses Gebietes zu veranschaufichen. Die Anwendung gemeinsamer Theorien mit anderen forensischen Wissenschaften macht dieses Teilgebiet der forensischen Informatik tatsächlich zu einer forensischen Wissenschaft.
2. Forensische Informatik ***im weiteren Sinne*** umfasst den gesamten Prozess

der Durchführung einer digitalen Ermittlung, wie beispielsweise die Suche nach digitalen Spuren, das Rekonstruieren gelöschter Daten, der Umgang mit großen Speichermengen und so weiter.

Der Bereich der forensischen Informatik im weiteren Sinne integriert viele Themenbereiche der Informatik, die bisher unabhängig voneinander agierten, wie zum Beispiel die Massendatenanalyse (***data mining, big data***), das Netzwerk-Fingerprinting, Reverse Engineering, die Hauptspeicheranalyse, Seitenkanalanalyse und Datenschutztechniken.

In der Praxis deutet sich eine thematische Aufteilung der eingesetzten Techniken an, die sich an der Flüchtigkeit der betrachteten digitalen Spuren orientiert (vergleiche Abschnitt 2.3.1). So kann man drei Unterbereiche dieses Gebietes unterscheiden:

1. Sicherung und Analyse persistenter Spuren, also vornehmlich Festplatten. In Anlehnung an die englischsprachige Terminologie wird dies oft als die ***Tot-Analyse*** (***dead analysis***) oder ***Post-Mortem-Analyse*** bezeichnet.
2. Sicherung und Analyse von flüchtigen Spuren im weiteren Sinne, also der Inhalte von Hauptspeicher, Caches etc. (zumeist im Rahmen einer ***Live-Analyse***).
3. Sicherung und Analyse von flüchtigen Spuren im engeren Sinne, meist Spuren im Netz. Dieser Teilbereich wird häufig als ***Netzwerkforensik*** bezeichnet.

2.5.3 Fragestellungen der forensischen Informatik im engeren Sinne

In Abschnitt 2.4 haben wir bereits gezeigt, dass man viele Ermittlungsfragestellungen auf Fragen nach Assoziationen reduzieren kann. Die strikte Anwendung der Theorie von Transfer und Assoziation zwingt den Forensiker, einzelne präzise Aussagen über seine Befunde zu formulieren. Da der Begriff des Transfers so grundlegend und bereits aus anderen forensischen Disziplinen bekannt ist, sind die Aussagen, welche am Ende des Prozesses zur Assoziation getroffen werden, auch für technisch meist weniger versierte Personen, wie beispielsweise Richter oder Rechtsanwälte, leicht verständlich. Die in Abschnitt 2.4 behandelten Beispiele untermauern dies. Die dort getroffenen Aussagen sind folgende:

1. Foto ***A*** wurde durch Digitalkamera ***B*** aufgenommen.
2. USB-Speichergerät ***A*** war an Computer ***B*** angeschlossen.
3. Computer ***A*** hat Website ***B*** aufgerufen.
4. Daten an Stelle ***A*** wurden von Speicherstelle ***B*** kopiert.

Staatsanwälte formulieren gerade solche Aussagen häufig als Fragen an die forensischen Wissenschaftler. Sie beruhen auf einzelnen und leicht nachvollziehbaren Assoziationen, aus welchen Staatsanwälte oder Richter Schlussfolgerungen ableiten können. Es gibt weit mehr Beispiele solcher Fragen, die im Kontext einer Ermittlung gestellt werden können. Hier folgen nur einige wenige Beispiele:

- Wurde Website ***A*** auf Rechner ***B*** aufgerufen?
- Wurde E-Mail ***A*** (Absender, Empfänger, Datum, Betreff, Inhalt, Datei-Anhänge) auf diesem Rechner verfasst und an Rechner ***B*** verschickt?
- War USB-Stick ***A*** einmal an Rechner ***B*** angeschlossen?
- Wurde Datei ***A*** (bitweiser Vergleich oder Hash-Vergleich) auf Rechner ***B*** kopiert oder heruntergeladen?
- Findet sich Stichwort ***A*** in Dokumenten auf Rechner ***B?***
- Befinden sich auf Rechner ***A*** Videos mit Inhalt ***B?***
- Wurde auf Rechner ***A*** Software ***B***, wie beispielsweise eine bestimmte Filesharingoder Datenvernichtungssoftware, eingesetzt?
- Wurde Rechner ***A*** durch Malware ***B***, wie beispielsweise einen Banken-Trojaner, kompromittiert?
- Wurde mit Rechner ***A*** eine Instant Messenging-/VoIP- Kommunikation mit folgender/m Person/Synonym ***B*** geführt?

Derartige Hypothesen zu belegen oder zu widerlegen stellt den Kern dessen dar, was forensische Wissenschaftler in der forensischen Informatik tun sollten. Alle Untersuchungen und Analysen sollten auf die Beantwortung solcher Fragen nach Assoziationen in der digitalen Welt ausgerichtet sein. Die Beantwortung jeder dieser Fragestellungen erfordert ein hohes Maß an Spezialisierung und Fachkenntnis, um das in Gerichtsverfahren geforderte Niveau an Gewissheit zu

erlangen. Dies ist der Teil der Computerforensik, der am deutlichsten in der Tradition anderer forensischer Wissenschaften steht.

Die Leitung einer Ermittlung und die Formulierung der durch forensische Wissenschaftler zu beantwortenden Fragen stellt den anderen Teil forensischen Informatik dar, der häufig ebenfalls als Computerforensik bezeichnet wird (Geschonneck, 2006) und durch entsprechend geschulte Polizeibeamte und Juristen durchgeführt werden kann: Diese Aufgabe erfordert ein gewisses Maß an Wissen über Verbrechen (Modus Operandi, Kriminologie usw.) und ein Basiswissen im Bereich der elektronischen Datenverarbeitung. Es besteht jedoch keine Notwendigkeit dafür, dass diese Personen forensische Wissenschaftler sein müssen und daher ein hohes Maß an technischer Expertise im Bereich der Informatik erwerben müssen. Hervorzuheben ist jedoch, dass beide Aspekte der forensischen Informatik gleichmaßen wichtig sind.

Die beschriebene Unterteilung der Computerforensik in diese beiden geschilderten Aspekte fördert eine wissenschaftlich ausgerichtete forensische Informatik und die Erreichung von Standards, wie sie in forensischen Wissenschaften anderer Bereiche bereits seit langem als normal angesehen werden.

2.5.4 Fragestellungen der forensischen Informatik im weiteren Sinne

Eine der wichtigsten Problemstellungen der forensischen Informatik im weiteren Sinne ist die Aufbereitung großer Datenmengen. Die dabei auftretenden Probleme sind außerordentlich vielfältig und aufwändig. Wir werden viele dieser Probleme in Kapitel 8 erörtern.

2.6 Verwandte Literatur

2.6.1 Zur Integration der Informatik in die Reihe forensischer Wissenschaften

Es gab bereits mehrere Ansätze, die praktische Computerforensik mit klassischen Theorien der Forensik zu vereinen. Vor allem ist hier die Arbeit von Carrier u. Spafford (2003) zu nennen, die den ***digitalen Tatort*** (***digital crime scene***) in Analogie zum physischen Tatort definieren. Allerdings basieren deren

Empfehlungen auf Büchern (James u. Nordby, 2009; Saferstein, 2010) (in älteren Auflagen), deren Fokus eher in der Erläuterung konkreter Techniken als in der Untersuchung einheitlicher Theorien besteht. Ebenso ordnen Carrier u. Spafford (2004) die Rekonstruktion von Ereignissen der digitalen Welt den entsprechenden Konzepten der physischen Welt zu. Das Konzept des Transfers klingt aber lediglich implizit bei der Betrachtung von Zustandsübergängen in Automaten an.

Pollitt (2008) war der erste, der die Theorie von Inman u. Rudin (2000) auf die Computerforensik anwandte. Pollitt identifiziert bereits die Notwendigkeit zur Anwendung aus der klassischen Forensik bekannter Theorien auf das Gebiet der Computerforensik. Er skizziert, wie Inmans und Rudins Prozess der Assoziation in der digitalen Welt aussehen könnte. Das, wie in Kapitel 1 geschildert, zugehörige Konzept des Transfers liegt jedoch außerhalb von Pollitts Fokus.

Weiterhin haben auch Cohen (2010, 2011) und Cohen u. a. (2011) die digitale Forensik in den Kontext von Inman u. Rudin (2000) gestellt. Deren Fokus liegt jedoch vielmehr auf dem Konzept des Transfers als auf dem der Assoziation und dem zugehörigen Prozess hin zur Assoziation. Eine vollständige Integration der forensischen Informatik in den Kontext der Theorie von Inman u. Rudin (2000) wurde erstmals von Dewald u. Freiling (2012) entwickelt (vgl. auch die erweiterte Version von Dewald u. Freiling (2014)).

2.6.2 Zur Praxis der Computerforensik

Es gibt eine große Anzahl von Basiswerken der Computerforensik, die sich primär technischen Aspekten der Durchführung von Untersuchungen widmen.

Kruse II u. Heiser (2001) beschreiben in ihrem Buch praxisnah die forensische Sicherung und Auswertung von Datenträgern unter Benutzung konkreter Werkzeuge. So werden die grundlegenden Funktionen dieser Software anhand von Bildschirmfotos oder der Aufrufsyntax von Kommandozeilen-basierten Anwendungen erläutert. Das Werk von Kruse II und Heiser beschreibt weiterhin die Untersuchung spezifischer Fall-Beispiele, gibt Ratschläge zur Analyse von Windows- und Unix-basierten Systemen und versteht sich als Handbuch für Ermittler im Bereich der Computerforensik.

Das Buch ***Forensic Discovery*** von Farmer u. Venema (2005) gehört neben Casey (2011) sicherlich zu den wichtigsten englischsprachigen Standardwerken. Im ersten Kapitel liefern die Autoren auch einen Überblick über einige der

wichtigsten praktischen Grundlagen, wie die ***Reihenfolge der Flüchtigkeit*** (***Order Of Volatility, OOV***), Abstraktionsebenen von Daten, die Manipulierbarkeit von Daten und die prinzipielle Wiederherstellbarkeit gelöschter Daten. Wirklich grundlegende Theorien sind in diesem eher praxisorientierten Leitfaden jedoch verständlicherweise nicht zu finden.

Pauf (2009) betrachtet digitale Spuren vor allem im Lichte amerikanischer Rechtsprechung und geht zunächst auf juristische Anforderungen an digitale Spuren ein. Hier nennt Pauf die in dieser Arbeit angesprochene wissenschaftliche Methode bei der Erhebung digitaler Spuren als Forderung des amerikanischen Verfassungsgerichtes (des ***supreme court***), welcher hierzu ebenfalls die Arbeit von Popper (1962) zitiert. Interessant sind weiterhin Paufs Ausführungen über Ereignisse (***events***) in digitalen Systemen, welche er als das grundlegendste Phänomen der digitalen Welt bezeichnet. Ereignisse — so Pauf — sind die Ursache für die Modifizierung von Daten und der Entstehung von Spuren. Er erkennt weiter, dass ein solches Ereignis in der digitalen Welt zumeist nicht wirklich aus einem einzelnen Ereignis, sondern aus einer Folge vieler Einzelereignisse besteht.

> „Naturally, events precede or follow one another in time. One event may cause one or more subsequent events.“ (Pauf, 2009, S. 27)

Diese Überlegungen sind denen dieser Arbeit im Ansatz durchaus ähnlich, leider bleiben dies jedoch die einzigen grundlegenden Betrachtungen in Paufs Buch.

Als deutschsprachiges Pendant zu Farmer u. Venema (2005) kann das Buch ***Computer Forensik*** von Geschonneck (2006) angesehen werden. Auch dieses Werk ist primär als pragmatisch geprägtes Handbuch für die Durchführung spezifischer digital-forensischer Untersuchungen zu sehen und behandelt beispielsweise den korrekten Umgang mit digitalen Beweismitteln, deren gerichtsverwertbare Sicherung, die Anfertigung eines Datenträgerabbildes oder die Wiederherstellung gelöschter Daten anhand konkreter Dateisysteme. Geschonneck gibt weiterhin einen Überblick und eine kurze Einführung in die gängigsten freien und kommerziellen Software-Werkzeuge. Eine Betrachtung theoretischer Grundlagen fehlt hier jedoch.

2.7 Zusammenfassung

In diesem Kapitel haben wir kurz die Entstehung digitaler Spuren betrachtet und ihre Eigenschaften untersucht. Daraufhin wurde die Herstellung von Assoziationen in der digitalen Welt als Kern der forensischen Informatik als forensische Wissenschaft identifiziert und mit der aktuellen Auffassung digitaler Forensik in der Praxis verglichen.

---

[1] http://www.mozilla.org/thunderbird/

Kapitel 3

Spuren, Interferenz und die Lösbarkeit von Rekonstruktionsproblemen

Autor: Andreas Dewald

In der forensischen Informatik stellt die Feststellung von Assoziationen und damit die Rekonstruktion von Ereignissen den Kern einer jeden Untersuchung dar. Im Gegensatz zu anderen forensischen Wissenschaften geht es in der forensischen Informatik um Hypothesen über Ereignisse in einem digitalen System und nicht in der physischen Welt. Die Spuren eines solchen digitalen Ereignisses liegen in Form digitaler Daten vor. Die grundsätzliche Fragestellung, mit der wir uns in diesem Kapitel im Detail beschäftigen, ist die nach den Umständen, unter denen es möglich ist, Ereignisse aus den Spuren eines digitalen Systems zu rekonstruieren, wie es die Theorie nach Inman u. Rudin (2000) fordert (siehe Kapitel 1).

Wie man schnell auf den folgenden Seiten sieht, erfolgt diese Betrachtung ***rein theoretisch*** in einem Modell digitaler Systeme. Es werden zwar auch Aussagen bewiesen, dies erfolgt aber im Sinne eines mathematischen Beweises (und nicht eines Beweises vor Gericht). Man wird auch schnell sehen, dass die Annahmen, unter denen die Zusammenhänge bewiesen werden, die Aussagekraft der Ergebnisse für die Praxis zum Teil stark einschränken. Neben der intellektuellen Herausforderung besteht der Reiz und der Nutzen dieses Kapitels jedoch im Aufstellen einer präzisen Terminologie, die auch unter weniger scharfen Randbedingungen sinnvoll und wenig missverständlich ist.

Wir befassen uns also mit ***digitalen Spuren***, die durch ***Ereignisse*** in einem ***digitalen System*** hervorgerufen werden. Bevor wir uns mit der Beantwortung der Frage befassen können, wann ein Ereignis in einem digitalen System Spuren hinterlässt, benötigen wir zunächst eine konkrete Vorstellung davon, wie ein solches digitales System modelliert werden kann, was Ereignisse in diesem System sind und wie sich darin digitale Spuren darstellen.

In diesem Kapitel führen wir daher zunächst ein abstraktes Modell für digitale Systeme ein. Im Kontext dieses Modells formulieren wir dann formal die konkreten Probleme, deren Lösbarkeit wir anhand des Modells später im Detail untersuchen. Weiterhin modellieren wir ***Spuren*** und Eigenschaften von Spuren, welche das Belegen von Hypothesen über vergangene Ereignisse innerhalb des digitalen Systems ermöglichen. Analog definieren wir ***Kontraspuren*** und deren Eigenschaften als Spuren für die Widerlegung vergangener Ereignisse. Anschließend betrachten wir Kriterien für die Entstehung und Auslöschung solcher Spuren und prüfen, was Kriterien für die Lösbarkeit der eingangs definierten Probleme sind.

Am Ende steht die Erkenntnis über innere Zusammenhänge und Strukturen digitaler Spuren auf Basis der eingeführten Formalisierung. Auch die Frage nach Voraussetzungen für die Rekonstruierbarkeit vergangener Ereignisse innerhalb eines digitalen Systems können am Ende des Kapitels beantwortet und bewiesen werden.

3.1 Intuition

Eine in der Informatik übliche und intuitive Modellierung eines digitalen Systems ist die des endlichen Zustandsautomaten (Hopcroft u. a., 2002). In einem solchen endlichen Automaten wird ein System $\boldsymbol{S}$ durch eine Menge von Zuständen $\boldsymbol{Q}$ und eine Menge von Zustandsübergangen Σ definiert. Diese Zustandsübergänge modellieren die gesamte Systemaktivität und können daher in forensischer Sichtweise als kleinstmögliche Ereignisse aufgefasst werden. Das System $\boldsymbol{S}$ befindet sich zu jeder Zeit in einem bestimmten Zustand $\boldsymbol{q} \in \boldsymbol{Q}$. Ein solcher Zustand repräsentiert alle in diesem System gespeicherten Daten zu diesem Zeitpunkt. Weiterhin wird der Initialzustand $\boldsymbol{q}_0$, sowie eine Menge von Endzuständen definiert. Diese Modellierung lässt sich, wie in Abbildung 3.1 beispielhaft dargestellt, in Form eines Graphen visualisieren.

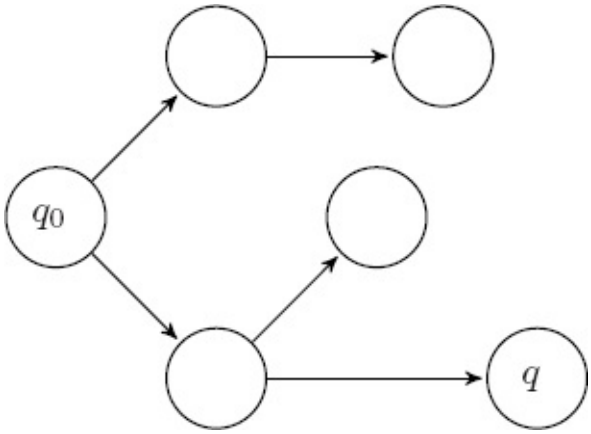


Abbildung 3.1: Beispielhafte Visualisierung eines Zustandsautomaten.

Legen wir ein solches System $\boldsymbol{S}$ in einem Zustand $\boldsymbol{q}$ zugrunde. Im Allgemeinen möchten wir nun diejenige Folge von Zustandsübergangen finden, die zum gegebenen Zustand $\boldsymbol{q}$ führte. Die Intuition ist, dass es sich bei dem System um einen forensisch zu analysierenden Computer handelt und bei dem Zustand $\boldsymbol{q}$ um eben jenen Zustand, in dem sich das zu analysierende System nun befinden, also beispielsweise nachdem damit eine Straftat begangen und das System sichergestellt wurde. Zur Beweisführung möchten wir nun rekonstruieren, welche Ereignisse stattgefunden haben. Hierzu ist es notwendig, diejenigen Sequenzen von Zustandsübergängen (Pfade im Graphen), welche vom initialen Zustand $\boldsymbol{q}_0$ zum gegebenen Zustand $\boldsymbol{q}$ führen, zu rekonstruieren. Dies wird in Abbildung 3.2 schematisch dargestellt.

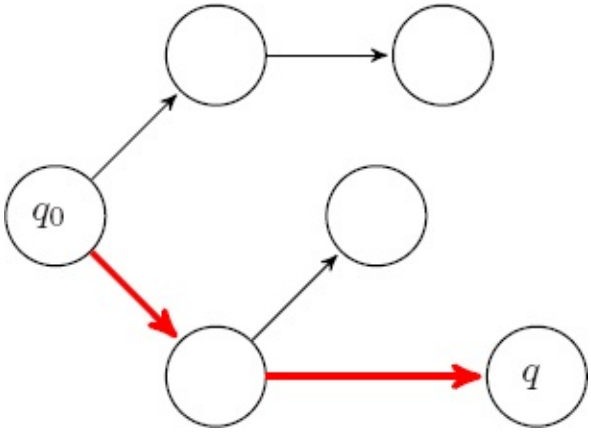


Abbildung 3.2: Beispielhafte Visualisierung des zu rekonstruierenden Pfades in einem Zustandsautomaten.

Obgleich hierzu die Kenntnis über das gesamte System notwendig ist, ist die Rekonstruktion aller Pfade, die zu einem gegebenen Zustand führen, prinzipiell möglich. Problematisch wird die Rekonstruktion jedoch, falls in einem System mehrere im gegebenen Zustand ***q*** resultierende Pfade existieren, wie es Abbildung 3.3 illustriert. Offensichtlich ist eine eindeutige Rekonstruktion der Ereignis-Historie eines Systems unter solchen Umständen nicht immer möglich. Um derartige Fälle genauer betrachten und insbesondere Grenzfälle der Rekonstruierbarkeit von Ereignissen in digitalen Systemen untersuchen zu können, führen wir im folgenden Abschnitt ein verfeinertes Systemmodell ein.

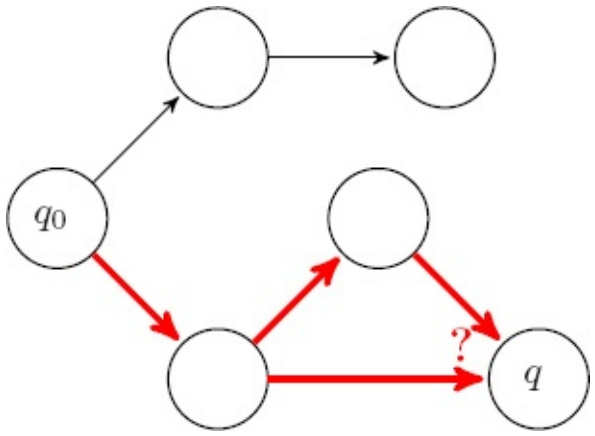


Abbildung 3.3: Beispielhafte Visualisierung alternativer Pfade in einem Zustandsautomaten.

3.2 Modell eines digitalen Systems

Wir führen nun ein verfeinertes Modell eines digitalen Systems ein, welches sich besser zur Modellierung der betrachteten Problemstellung eignet als das im letzten Abschnitt beschriebene intuitive Modell. Wir nutzen hierzu Dijkstras ***Guarded Commands*** (Dijkstra, 1975) als bekannten und gut untersuchten Formalismus. Dijkstras Notation von Programmen findet in ähnlicher Weise auch in ***Unity*** (Chandy u. Misra, 1988) Verwendung. Insbesondere kann mit Hilfe von ***Guarded Commands*** auch Nebenläufigkeit modelliert werden. Obgleich dies zunächst keine notwendige Voraussetzung für die nachfolgenden Betrachtungen darstellt, behalten unsere Ergebnisse auch in nebenläufigen Systemen ihre Gültigkeit. Damit ist die verwendete Notation sehr gut geeignet, reale Computersysteme abzubilden.

Dijkstras Modell basiert auf der Vorstellung, dass ein Computer über eine endliche Anzahl von Speicherzellen verfügt, in deren Inhalt sich der Zustand des Systems manifestiert. Diese Vorstellung wird bei Dijkstra (1975) formal durch eine Menge von Variablen ausgedrückt, wie wir im folgenden Abschnitt erläutern.

3.2.1 Variablen

Wir modellieren den Zustand eines digitalen Systems als eine Menge von ***Variablen*** (***Variables***) $\boldsymbol{V}$. Variablen sind Container, die jeweils einen Wert speichern können. Intuitiv können Variablen Dateisystemobjekte wie Dateien und Verzeichnisse, aber auch Festplattenblöcke oder Hauptspeicherregionen repräsentieren. Sie können jedoch ebenso Attribute solcher Objekte darstellen, wie beispielsweise den Zeitstempel der letzten Modifikation einer Datei oder die Zugriffsrechte eines Verzeichnisses. Jede Variable besitzt eine endliche ***Domäne*** (***Domain***) $\boldsymbol{D}$, die definiert, welche Art von Werten in dieser Variable gespeichert werden können.

In realen Computersystemen besitzen (zumindest auf Bitebene) sogar alle Variablen dieselbe binäre Domäne, nämlich die Menge $\boldsymbol{D} = \{0,1\}$. Wir werden daher hier meist Beispiele über Variablen mit binärer Domäne anführen.

Ein ***Zustand*** (***State***) eines digitalen Systems ist nun eine Belegung aller Variablen in $\boldsymbol{V}$ mit konkreten Werten. Wir drücken Zustände als Mengen von Variable-Wert-Paaren $\boldsymbol{v} = \boldsymbol{d}$ mit $\boldsymbol{v} \in \boldsymbol{V}$ und $\boldsymbol{d} \in \boldsymbol{D}$ aus. Der besseren Lesbarkeit halber setzen wir solche Zuweisungen, wo sinnvoll, in Klammern und schreiben $[\boldsymbol{v} = \boldsymbol{d}]$, um zu verdeutlichen, dass es sich hierbei um die Belegung einer Variablen handelt. Wenn wir als einfaches Beispiel $\boldsymbol{V}$ definieren als $\boldsymbol{V} = \{\boldsymbol{a}, \boldsymbol{b}\}$, dann ist $\boldsymbol{q} = \{\boldsymbol{a} = 0, \boldsymbol{b} = 1\}$ ein Zustand. Mit $\boldsymbol{Q}$ bezeichnen wir die Menge aller hinsichtlich $\boldsymbol{V}$ möglichen Zustände. In unserem Beispiel wäre:

$$\boldsymbol{Q} = \{\{\boldsymbol{a} = 0, \boldsymbol{b} = 0\}, \{\boldsymbol{a} = 0, \boldsymbol{b} = 1\}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 0\}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 1\}\}$$

Wir nennen die initiale Belegung aller Variablen des Systems den ***Initialzustand*** (***Initial State***) $\boldsymbol{q}_0 \in \boldsymbol{Q}$. Weiterhin nennen wir die initiale Belegung einer konkreten Variablen $\boldsymbol{v} \in \boldsymbol{V}$ den ***Initialwert*** (***Initial Value***) dieser Variablen und bezeichnen diesen als $\boldsymbol{IV}(\boldsymbol{v})$. Formal gilt:

$$\boldsymbol{IV}(\boldsymbol{v}) := \boldsymbol{d} \Leftrightarrow [\boldsymbol{v} = \boldsymbol{d}] \in \boldsymbol{q}_0 \qquad (3.1)$$

3.2.2 Programme

Gegeben sei eine Menge von Variablen ***V*** wie beschrieben. Wir modellieren nun Aktivitäten eines Computersystems durch den Begriff des Programmes wie folgt: Ein ***Programm*** (***Program***) ***S*** besteht aus einer endlichen Menge von Aktionen Σ. Eine ***Aktion*** (***Action***) ***σ*** ist ein Tupel (***g***, ***c***), bestehend aus einem ***Wächter*** (***Guard***) ***g*** und einem ***Befehl*** (***Command***) ***c***. Der ***Wächter*** ist ein boolescher Ausdruck über Variablen in ***V*** , und der ***Befehl*** ist eine Zuweisung von Werten zu Variablen in ***V*** . Um auszudrücken, dass eine konkrete Zuweisung [***v*** = ***d***] (***v*** ∈ ***V***, ***d*** ∈ ***D***) von einer Aktion ***σ*** ∈ Σ durchgeführt wird, schreiben wir der Übersichtlichkeit halber [***v*** = ***d***] ∈ ***σ*** anstatt [***v*** = ***d***] ∈ ***c*** für ***σ*** = (***g***, ***c***). Wir bezeichnen die Menge von Variablen, denen eine Aktion ***σ*** Werte zuweist, mit ***vars***(***σ***):

$$\boldsymbol{vars}\,(\boldsymbol{\sigma}) = \{\boldsymbol{v} \in \boldsymbol{V} \mid \exists \boldsymbol{d} \in \boldsymbol{D} : [\boldsymbol{v} = \boldsymbol{d}] \in \boldsymbol{\sigma}\} \tag{3.2}$$

Da im Initialzustand die Zuweisung des Initialwertes zu einer Variablen offensichtlich keine änderung an der Variablen hervorruft, müssen Zuweisungen von Initialwerten an einigen Stellen besonders beachtet werden. Aus diesem Grund definieren wir als Einschränkung der Variablenmenge einer Aktion ***σ*** auch die Menge nur derjenigen Variablen, denen ***σ*** einen ***vom Initialwert verschiedenen*** Wert zuweist und damit tatsächlich den Wert der Variablen im Initialzustand ***verändert***. Wir definieren diese Menge als von ***σ veränderte Variablen*** (***changed variables***) ***cvars***(***σ***) wie folgt:

$$cvars(\sigma) = \{v \in V \mid \exists d \in D : [v = d] \in \sigma \wedge d \neq IV(v)\} \tag{3.3}$$

Es gilt damit per Definition für alle Aktionen ***σ*** ∈ Σ, dass ***cvars***(***σ***) eine Teilmenge der Variablenmenge ***vars***(***σ***) ist und wenn ***σ*** keine Zuweisungen von Initialwerten vornimmt gilt: ***cvars***(***σ***) = ***vars***(***σ***).

Variablen: {***a***, ***b***}
Initialzustand: {***a*** = 0, ***b*** =0}
Aktionen:

`p1aktion1`: ***a*** = 0 → ***a*** := 1
`p1aktion2`: ***b*** = 0 → ***b*** := 1

Abbildung 3.4: ***Programm 1***.

Ein System $\boldsymbol{S}$ (beschrieben durch ein Programm) wird also zusammenfassend definiert als $\boldsymbol{S} = (\boldsymbol{V}, \Sigma, \boldsymbol{q}_0)$. Im Folgenden benutzen wir immer diese Notation und bezeichnen beispielsweise mit Σ die Menge aller Aktionen im System $\boldsymbol{S}$, wie hier definiert.

Beispiel 9 (Einfaches Programm in bewachten Anweisungen) ***Als Beispiel soll Abbildung 3.4 die verwendete Notation von Programmen darstellen: Unter Variablen werden alle Variablen in V benannt, und unter Initialzustand wird deren initiale Belegung angegeben. Diese initialen Werte aller Variablen definieren den Startzustand $q_0 \in Q$ des Computersystems. Unter Aktionen werden schließlich alle Aktionen definiert. Die Definition einer Aktion beginnt hierbei mit dem Namen der Aktion (in diesem Beispiel `p1aktion1` und `p1aktion2`) und einem Doppelpunkt, gefolgt von Wächter und Befehl. Wächter und Befehl werden durch ein Pfeilsymbol getrennt***.

Gegeben sei ein konkreter Zustand $\boldsymbol{q}$. Dann nennen wir eine Aktion $(\boldsymbol{g}, \boldsymbol{c})$ ***aktiv in q***, genau dann, wenn der Guard $\boldsymbol{g}$ in Zustand $\boldsymbol{q}$ ***wahr*** ist. Intuitiv arbeitet das System sein Programm wie folgt ab: Ausgehend von $\boldsymbol{q}_0$ wird eine beliebige aktive bewachte Anweisung ausgewählt und deren Befehl ausgeführt. Der resultierende Zustand ist dann Gegenstand einer weiteren Auswahlrunde. In dieser Weise setzt sich die Ausführung fort.

Auch wenn die Rolle der Guards in den folgenden einfach gehaltenen Beispielen zunächst untergehen mag, so ist deren Bedeutung für das Modell doch essenziell, da sie die Programmlogik abbilden. Beispielsweise kann so gezielt eine bestimmte Abfolge von Aktionen erzwungen werden oder die mehrfache Ausführung einer Aktion unterbunden werden. Im Wesentlichen schränken Wächter die Zustandsübergangsrelation ein und ermöglichen so erst eine sinnvolle Kontrolle des Systemverhaltens.

3.2.3 Pfade

Sei $\boldsymbol{S} = (\boldsymbol{V}, \Sigma, \boldsymbol{q}_0)$ ein System. Ein ***Pfad*** (***Trace***) $\boldsymbol{\alpha}$ von $\boldsymbol{S}$ ist eine alternierende Sequenz

$$\alpha = \langle q_0, \sigma_0, q_1, \sigma_1, q_2, \sigma_2, \ldots \rangle$$

von Zuständen und Aktionen von S, so dass q_0 der erste Zustand in dieser Sequenz ist und für alle $i \geq 0$ gilt, dass q_{i+1} durch die Ausführung der aktiven Aktion σ_i im Zustand q_i erreicht wird.

Beispiel 10 (Pfade) ***Die in Abbildung 3.5 dargestellten Sequenzen sind beispielsweise Pfade des in Abbildung 3.4 auf der vorherigen Seite definierten Beispielprogramms*** Programm 1.

$\langle \{a = 0, b = 0\} \rangle$,
$\langle \{a = 0, b = 0\}$,`p1aktion1`, $\{a = 1, b = 0\} \rangle$,
$\langle \{a = 0, b = 0\}$,`p1aktion2`, $\{a = 0, b = 0\} \rangle$,
$\langle \{a = 0, b = 0\}$,`p1aktion1`, $\{a = 1, b = 0\}$,`p1aktion2`, $\{a = 1, b = 1\} \rangle$,
$\langle \{a = 0, b = 0\}$,`p1aktion2`, $\{a = 0, b = 1\}$,`p1aktion1`, $\{a = 1, b = 1\} \rangle$

Abbildung 3.5: Pfade von ***Programm 1***.

Ein Pfad ist ***endlich*** (***finite***), wenn die Länge der Sequenz endlich ist. Wir sagen, dass endliche Pfade in einem Zustand enden. Einen solchen Zustand nennen wir den ***Endzustand*** (***Final State***) dieses Pfades. Wir nennen einen endlichen Pfad ***maximal***, wenn in seinem Endzustand keine Aktion aktiv ist, das heißt, wenn keine Aktion mehr aus dem Zustand herausführt. Im Allgemeinen müssen Pfade jedoch nicht maximal sein, da sie in jedem erreichbaren Zustand enden können. Für jeden endlichen Pfad α bezeichnet ***final***(α) den letzten Zustand von α. Das heißt für $\alpha = \langle q_0, \sigma_0, \ldots, \sigma_{n-1}, q_n \rangle$ ist ***final***(α) = q_n. Falls α ein Präfix eines anderen Pfades α' ist, schreiben wir $\alpha \sqsubseteq \alpha'$.

Die Menge aller möglichen Pfade eines Systems nennen wir A. Die Menge aller Pfade α, die in einem gegebenen Zustand $q \in Q$ enden, bezeichnen wir mit $A(q)$. Formal:

$$\mathcal{A}(q) = \{\alpha \in A \mid final(\alpha) = q\} \tag{3.4}$$

Es ist leicht zu sehen, dass für alle $q \in Q$ gilt: $A(q) \subseteq A$.

Falls $A(q)$ gleich der leeren Menge ist ($A(q) = \varnothing$), so ist q ein nicht erreichbarer Zustand in diesem System. Das bedeutet, dass das System ohne äußere Einflüsse oder Manipulation nie in diesem Zustand vorgefunden werden

kann. Würde das System dennoch in einem nicht erreichbaren Zustand vorgefunden, so bestünde die Aufgabe einer Untersuchung darin, die ***wahrscheinlichste*** Ereignis-Historie des Systems zu ermitteln, also diejenigen Pfade $\boldsymbol{\alpha'}$, deren Endzustände $\boldsymbol{q'} = \boldsymbol{final}(\boldsymbol{\alpha'})$ beispielsweise die geringste Abweichung von Zustand $\boldsymbol{q}$ aufweisen, wenn anzunehmen ist, dass das System nur zu einem geringen Teil modifiziert wurde. Diese Problemstellung weist Ähnlichkeiten mit dem Bereich der fehlertoleranten Programmierung (Gärtner u. Völzer, 2000) oder auch fehlerkorrigierende Codes (Huffman u. Pless, 2003) auf. Eine echte Rekonstruktion vergangener Ereignisse ist jedoch nicht möglich. Aus diesem Grund werden wir in unseren weiteren Betrachtungen annehmen, dass $\boldsymbol{A}(\boldsymbol{q}) \neq \emptyset$ für alle $\boldsymbol{q} \in \boldsymbol{Q}$ gilt.

3.2.4 Nichtdeterminismus

Wenn in einem System in einem einzelnen Zustand mehr als eine Aktion aktiv ist, fällt die Entscheidung, welche der Aktionen ausgeführt werden soll, nicht-deterministisch. Würden diese Entscheidungen deterministisch getroffen, so würde dies die Rekonstruktion von Ereignissen vereinfachen. Dies ist jedoch für unsere Betrachtungen nicht notwendig, und die Annahme der Nicht-Determiniertheit macht den Formalismus realistischer.

Die Menge der Zustände und die möglichen Übergänge zwischen den Zuständen durch Ausführung von Aktionen können ähnlich wie die anfangs betrachteten endlichen Automaten als Zustandsübergangsdiagramm dargestellt werden. Abbildung 3.6 auf der nächsten Seite zeigt ein solches Diagramm für ***Programm 1***. Die Kreise stellen hierbei die (erreichbaren) Zustände des Systems mit den Werten aller Variablen in diesem Zustand dar. Die Pfeile zeigen mögliche Zustandsübergänge zwischen den Zuständen und sind mit den Namen der jeweiligen Aktionen gekennzeichnet.

Beispiel 11 (Aktive Anweisungen und maximale Pfade) ***Aus der Abbildung 3.6 wird ersichtlich, dass im Initialzustand beide Aktionen aktiv sind. Sobald eine der Aktionen ausgeführt wurde, ist sie im darauffolgenden Zustand inaktiv und nur die andere Aktion kann ausgeführt werden. Dies ist jedoch in diesem Beispiel absichtlich in dieser Weise konstruiert und keine allgemeine Eigenschaft des Modells. Im Zustand*** $\{\boldsymbol{a} = 1, \boldsymbol{b} = 1\}$ ***ist keine Aktion aktiv. Daher besitzt dieses Programm genau die beiden folgenden unterschiedlichen maximalen Pfade:***

$$\langle \{\boldsymbol{a} = 0, \boldsymbol{b} = 0\}, \texttt{p1aktion1}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 0\}, \texttt{p1aktion2}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 1\} \rangle$$

und

$$\langle \{\boldsymbol{a} = 0, \boldsymbol{b} = 0\}, \texttt{p1aktion2e}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 0\}, \texttt{p1aktion1}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 1\} \rangle.$$

3.2.5 Aktionen und Operationen

In realen Computersystemen werden primitive Aktionen häufig zu abstrakteren Einheiten gruppiert. Beispielsweise umfasst das Löschen einer E-Mail in einer E-Mail-Anwendung bereits auf Betriebssystemebene sehr viele einzelne Aktionen (Betriebssystem-API-Aufrufe) für das Öffnen von Dateien oder dem Aufbau einer Netzwerkverbindung. Auf Hardwareebene beinhaltet die Ausführung jedes solchen Aufrufes gar ein Vielfaches atomarer Maschineninstruktionen und Speicheroperationen. Wir bezeichnen daher solche Zusammenfassungen von Sequenzen beliebig vieler Aktionen zu einer semantischen Entität als ***Operationen***. Weitere Beispiele solcher Operationen sind das Starten einer konkreten Anwendung, das Versenden einer E-Mail oder das Öffnen einer Website mit dem Browser.

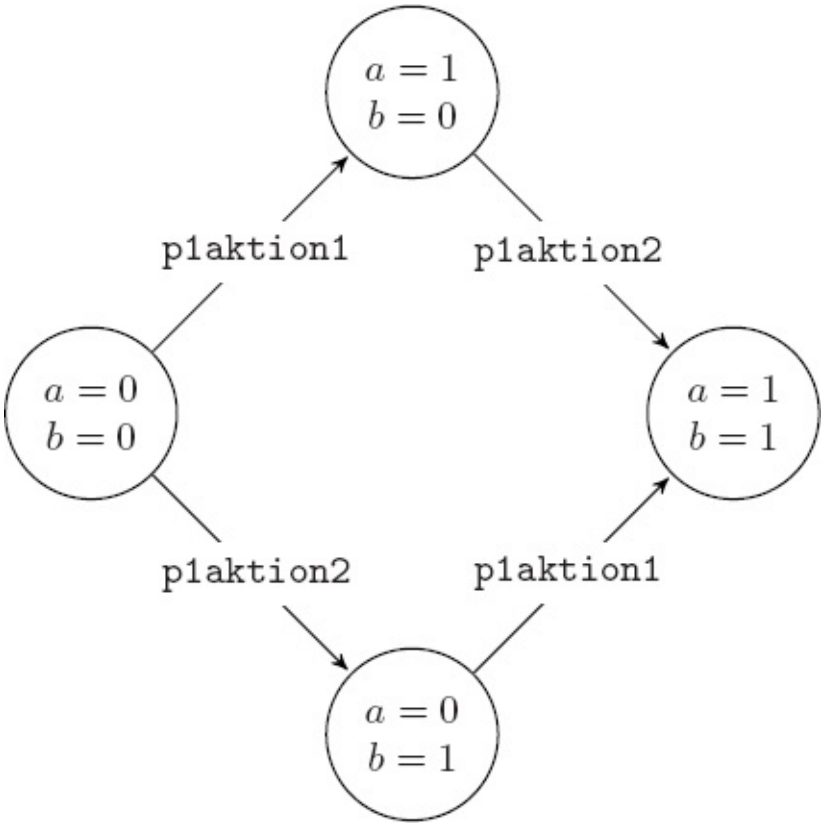


Abbildung 3.6: Zustandsübergangsdiagramm von ***Programm 1***.

3.2.6 Weitere Annahmen

Schließlich treffen wir noch ein paar zum Teil vereinfachende Annahmen:

- Wir beschränken uns auf Aktionen, die Variablen stets feste Werte zuweisen. Das bedeutet, dass wir beispielsweise ***Zähloperationen*** der Art $\boldsymbol{v} := \boldsymbol{v} + 1$ von unseren Betrachtungen ausschließen. Dies vereinfacht unsere Argumentation, schränkt aber die Allgemeingültigkeit der Ergebnisse nicht ein, da solche Zähloperationen aufgrund der Endlichkeit der Domänen durch eine Menge von Aktionen, welche feste Werte zuweisen, simuliert werden können. Zusätzlich erlauben solche Zähloperationen tiefere Rückschlüsse über einen Pfad, wie beispielsweise die Anzahl der Ausführungen einer solchen Aktion aufgrund des aktuellen Wertes der Zählvariablen. Können auf Basis des hier verwendeten Modells Rückschlüsse gezogen werden, so sind diese ebenso in Modellen mit Zähloperationen möglich.
- Wir nehmen an, dass alle Aktionen mindestens einer Variablen einen Wert zuweisen. Dies bedeutet, dass wir leere Aktionen außer Acht lassen, da sie offensichtlich keinerlei Auswirkung auf das System haben. Formal nehmen wir also an, dass für alle $\boldsymbol{\sigma} \in \Sigma$: $\boldsymbol{vars}(\boldsymbol{\sigma}) \neq \emptyset$ gilt.
- Wir nehmen an, dass alle Aktionen in einem System $\boldsymbol{S}$ ausgeführt werden können. Aktionen, die auf keinem Pfad des Systems ausgeführt werden können, haben offensichtlich keinerlei Auswirkung auf das Verhalten des Systems und werden daher von $\boldsymbol{S}$ ausgeschlossen.
- Computersysteme führen üblichweise nicht nur ein einzelnes Programm aus, sondern eine Vielzahl unterschiedlicher Programme inklusive eines Betriebssystems. Daher müssten wir, um das Gesamtverhalten eines Computers zu definieren, die Vereinigung der Mengen aller Variablen und Aktionen aller Programme bilden. Das Resultat wäre jedoch wiederum ein Programm (in unserer Notation), bestehend aus einer Menge von Variablen und Aktionen. Aus diesem Grund legt der Rest dieses Kapitels das hier eingeführte Modell eines Programms zugrunde. Es ist hervorzuheben, dass hier ein ***Programm*** stets auch eine solch abstrakte Modellierung eines gesamten Computersystems darstellen kann und nicht ausschließlich das,

was wir gemeinhin unter einem einzelnen Programm verstehen. Wir verwenden daher in diesem Kapitel weiterhin (wie bereits geschehen) die Begriffe ***Programm*** und ***System*** synonym.

Mit dieser abstrakten Modellierung lassen sich nun verschiedenste Phänomene auf allen Abstraktionsebenen eines realen Computers betrachten. Untersuchen wir beispielsweise Dateisystem-Zeitstempel, die in der Praxis für die Rekonstruktion eines Tathergangs häufig eine wichtige Rolle spielen, so können wir diese als Mengen von Variablen modellieren. Das NTFS-Dateisystem verwaltet beispielsweise für jede Datei vier 64-Bit Zeitstempel (Carrier, 2005), so dass wir einen solchen durch 64 binäre Variablen vollständig implementieren können. Als Aktionen können wir dann solche System-Aktivitäten auffassen, die zu Veränderungen der Zeitstempel führen. Analog lassen sich weitere häufig relevante Dateisystem-Attribute, wie Besitzerinformationen und Zugriffsrechte einer Datei, modellieren. Selbst der Inhalt beliebiger Dateien lässt sich als Menge von Variablen ausdrücken, da der verfügbare Speicherplatz eines realen Computers – trotz der enormen Größe heutiger Datenträger – immer endlich ist. Die Modellierung ist jedoch nicht auf die Ebene des Dateisystems beschränkt: Einzelne Bits des Hauptspeichers oder eines Datenträgers auf der Hardwareebene lassen sich genauso untersuchen, wie Felder der internen Struktur eines bestimmten Dateiformates auf Anwendungsebene. Unser Modell ist also durch seinen abstrakten Charakter sehr flexibel auf unterschiedliche Aspekte realer Computer anwendbar. Allerdings muss ein reales System durch den forensischen Informatiker stets im Einzelfall auf das Modell übertragen werden. Doch bevor wir den Blick auf die Übertragung des Theoretischen ins Praktische richten, untersuchen wir, welche Vorteile die formale Betrachtung eines Systems bietet. Wir formulieren hierzu im nächsten Abschnitt genau unsere Problemstellung und prüfen dann, welche Aussagen über die Lösbarkeit dieser Probleme wir formal zeigen können und welche inneren Zusammenhänge mit digitalen Spuren existieren.

3.3 Problemdefinition

Die Feststellung von Assoziationen bilden den Kern der forensischen Informatik im Sinne von Inman und Rudin. Im formalen Zusammenhang bezeichnen wir die Möglichkeit der Feststellung einer Assoziation kurz auch als

Rekonstruierbarkeit. Wir formalisieren die Rekonstruierbarkeit in drei Problemen im Kontext des eingeführten Systemmodells.

3.3.1 Allgemeines Rekonstruktionsproblem

Mit Hilfe des zuvor eingeführten Modells formulieren wir unsere ursprüngliche Fragestellung wie folgt: Sei $\boldsymbol{q}$ ein konkreter Zustand. Dann möchten wir alle Pfade des Systems rekonstruieren, die in der Vergangenheit ausgeführt worden sein könnten und das System in den gegebenen Zustand $\boldsymbol{q}$ geführt haben könnten. Wir definieren dieses Problem als das ***allgemeine Rekonstruktionsproblem***.

Definition 4 (Allgemeines Rekonstruktionsproblem) ***Sei*** $\boldsymbol{S} = (\boldsymbol{V}, \Sigma, \boldsymbol{q}_0)$ ***ein System. Das*** allgemeine Rekonstruktionsproblem (***General Reconstruction Problem***, GRP) ***für S und einen Zustand q ist die Rekonstruktion aller Pfade α des Systems, die in Zustand q führen. Gesucht ist also die Menge*** $\boldsymbol{A}(\boldsymbol{q})$.

Beispiel 12 (Allgemeines Rekonstruktionsproblem bei Programm 1) ***Im Beispiel von*** Programm 1 ***aus Abbildung 3.4 ist das allgemeine Rekonstruktionsproblem einfach zu lösen: Anhand von Abbildung 3.6 auf Seite → wird klar, dass die Variablen a und b „aufzeichnen“, ob Aktion*** `p1aktion1` ***und*** `p1aktion2` ***ausgeführt wurden. Gegeben den Zustand*** $\{\boldsymbol{a} = 0, \boldsymbol{b} = 1\}$, ***ist klar, dass auf allen zu diesem Zustand führenden Pfaden die Aktion*** `p1aktion1` ***nicht ausgeführt wird und Aktion*** `p1aktion2` ***definitiv ausgeführt wurde. In Zustand*** $\{\boldsymbol{a} = 1, \boldsymbol{b} = 1\}$ ***wissen wir, dass beide Aktionen*** `p1aktion1` ***und*** `p1aktion2` ***ausgeführt wurden, selbst wenn die Reihenfolge der Ausführung nicht eindeutig bestimmt werden kann***.

3.3.2 Spezifisches Rekonstruktionsproblem

In der Praxis ist es häufig gar nicht nötig, ***alle*** Pfade zu rekonstruieren, die zu einem gegebenen Zustand geführt haben könnten. Meist stellt sich vielmehr lediglich die Frage, ob ***eine bestimmte*** Aktion $\boldsymbol{\sigma}$ in der Vergangenheit stattgefunden hat. Etwas spezifischer ausgedrückt, gibt es drei unterschiedliche Fälle bezüglich einer Aktion $\boldsymbol{\sigma}$, und falls $\boldsymbol{A}(\boldsymbol{q})$ nicht leer ist, fällt $\boldsymbol{\sigma}$ auch immer in genau eine dieser Kategorien:

1. Um q zu erreichen, hat σ ***definitiv stattgefunden***.
   Formal: $\forall \alpha \in A(q) : \sigma \in \alpha$.
2. Um q zu erreichen, hat σ ***definitiv nicht stattgefunden***.
   Formal: $\forall \alpha \in A(q) : \sigma \notin \alpha$.
3. Um q zu erreichen, hat σ ***möglicherweise stattgefunden***.
   Formal: $\exists \alpha, \alpha' \in A(q) : (\sigma \in \alpha) \wedge (\sigma \notin \alpha')$.

Definition 5 (Spezifisches Rekonstruktionsproblem) ***Sei*** $S = (V, \Sigma, q_0)$ ***ein System. Das Spezifische Rekonstruktionsproblem (Specific Reconstruction Problem***, SRP) ***für*** S***, einen Zustand*** q ***und eine konkrete Aktion*** $\sigma \in \Sigma$ ***definieren wir als die Entscheidung, ob*** σ definitiv stattgefunden hat ***oder*** definitiv nicht stattgefunden hat, ***um Zustand*** q ***zu erreichen. Falls*** σ ***in die dritte Kategorie fällt, bezeichnen wir das SRP als nicht lösbar***.

Abbildung 3.7 veranschaufticht die drei Kategorien des Spezifischen Rekonstruktionsproblems und zeigt schematisch, dass Kategorie 3 die Konjunktion der Negationen der Kategorien 1 und 2 ist. Damit werden durch diese drei Kategorien alle Fälle vollständig abgedeckt.

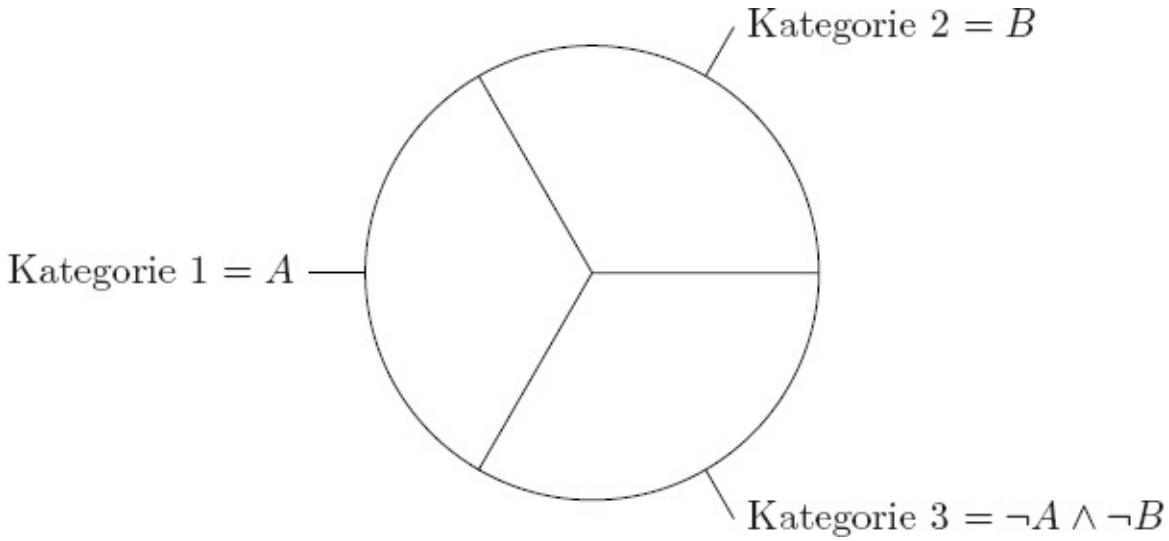


Abbildung 3.7: Visualisierung der drei orthogonalen Kategorien des spezifischen Rekonstruktionsproblems.

Beispiel 13 (Spezifisches Rekonstruktionsproblem bei Programm 1) ***Im Beispiel von*** Programm 1 ***aus Abbildung 3.4 können wir unter anderem folgende***

Aussagen hinsichtlich des spezifischen Rekonstruktionsproblems machen:

- ***`p1aktion1` hat definitiv vor Zustand*** $\{\boldsymbol{a} = 1, \boldsymbol{b} = 1\}$ ***stattgefunden***.
- ***`p1aktion2` hat definitiv vor Zustand*** $\{\boldsymbol{a} = 1, \boldsymbol{b} = 1\}$ ***stattgefunden***.
- ***`p1aktion1` hat definitiv nicht vor Zustand*** $\{\boldsymbol{a} = 0, \boldsymbol{b} = 1\}$ ***stattgefunden***.
- ***`p1aktion2` hat definitiv vor Zustand*** $\{\boldsymbol{a} = 0, \boldsymbol{b} = 1\}$ ***stattgefunden***.

3.3.3 Spezifisches Gruppen-Rekonstruktionsproblem

In einigen Fällen ist es sogar ausreichend zu wissen, ob zumindest eine Aktion einer bestimmten Menge von Aktionen stattgefunden hat oder nicht. Für eine solche Menge von Aktionen Σ' existieren wiederum drei unterschiedliche Fälle, analog zu denen des spezifischen Rekonstruktionsproblems:

1. Mindestens eine Aktion aus Σ' hat ***definitiv stattgefunden***, um $\boldsymbol{q}$ zu erreichen.
   Formal: $\forall \boldsymbol{\alpha} \in \boldsymbol{A}(\boldsymbol{q}) : \exists \boldsymbol{\sigma} \in \Sigma' : \boldsymbol{\sigma} \in \boldsymbol{\alpha}$.
2. Es hat ***definitiv keine*** Aktion aus Σ' stattgefunden, um $\boldsymbol{q}$ zu erreichen.
   Formal: $\forall \boldsymbol{\alpha} \in \boldsymbol{A}(\boldsymbol{q}) : \forall \boldsymbol{\sigma} \in \Sigma' : \boldsymbol{\sigma} \notin \boldsymbol{\alpha}$.
3. Eine oder mehrere Aktionen aus Σ' haben ***möglicherweise stattgefunden***, um $\boldsymbol{q}$ zu erreichen.
   Formal: $\exists \boldsymbol{\alpha}, \boldsymbol{\alpha'} \in \boldsymbol{A}(\boldsymbol{q}) : \exists \boldsymbol{\sigma}, \boldsymbol{\sigma'} \in \Sigma' : (\boldsymbol{\sigma} \in \boldsymbol{\alpha}) \wedge (\boldsymbol{\sigma'} \notin \boldsymbol{\alpha'})$

Der dritte Fall stellt wiederum die Konjunktion der Negationen der beiden anderen Fälle dar und die Fallunterscheidung ist damit vollständig.

Definition 6 (Spezifisches Gruppen-Rekonstruktionsproblem) ***Das*** Spezifische Gruppenrekonstruktionsproblem (***Specific Group Reconstruction Problem***, SGRP) ***für***

- ***ein System S (mit seiner Menge von Aktionen*** Σ***, der Menge von Variablen V und dem Initialzustand*** q_0),
- ***einen Zustand q und***
- ***eine konkrete Menge von Aktionen*** $\Sigma' \subseteq \Sigma$

definieren wir als die Entscheidung, ob Σ' ***in die erste oder zweite Kategorie fällt. Falls*** Σ' ***in die dritte Kategorie fällt, bezeichnen wir das SGRP als nicht lösbar***.

Beispiel 14 (Das SRP bei Programm 2) ***Als Beispiel betrachten wir Programm 2 aus Abbildung 3.8 auf der nächsten Seite mit Zustandsübergangsdiagramm in Abbildung 3.9. Hier ist das SRP sowohl für Aktion*** `p2aktion1` ***als auch für*** `p2aktion2` ***(abgesehen vom Initialzustand) nicht lösbar, da in Zustand*** $\{a = 1, b = 0\}$ ***nicht entschieden werden kann, ob die betreffende Aktion stattgefunden hat oder nicht. Jedoch lässt sich das SGRP für*** $\Sigma' = \{$`p2aktion1`, `p2aktion2`$\}$ ***lösen, da in Zustand*** $\{a = 1, b = 1\}$ ***klar ist, dass definitiv keine der beiden Aktionen aus*** Σ' ***stattgefunden hat und in Zustand*** $\{a = 1, b = 0\}$ ***mindestens eine wenn nicht gar beide Aktionen stattgefunden haben***.

Bei der Definition des SGRP wurde der Fall, dass die Menge Σ' gleich der Menge aller Aktionen des Systems Σ ist, nicht ausgeschlossen. Allerdings ist dieser Fall offensichtlich trivial, da sich für $\Sigma' = \Sigma$ das SGRP für jedes System lösen lässt. In diesem Fall liefert die Lösbarkeit des SGRP jedoch keine nützlichen Informationen über die Historie des Systems, da dann laut SGRP entweder mindestens eine Aktion aller Aktionen des Systems oder gar keine Aktion ausgeführt wurde. Diese Aussage ist offensichtlich gleichfalls trivial. Dieser Grenzfall zeigt bereits, dass die Lösbarkeit des SGRP umso weniger hilfreich ist, je größer die Menge Σ' ist. Es gilt also auch die minimale Menge Σ' zu finden, für die das SGRP lösbar ist, worauf wir später im Detail eingehen werden.

Nachdem wir nun unsere ursprüngliche Fragestellung in drei konkreten Problemstellungen formalisiert haben, widmen wir uns in den folgenden Abschnitten der Frage, unter welchen Umständen es möglich ist, diese Probleme zu lösen.

3.4 Spuren

Intuitiv hinterlassen Aktionen dann unterscheidbare Spuren, wenn es möglich ist, aus dem Zustand eines System zu schlussfolgern, welche Aktionen stattgefunden haben. Wir definieren nun den Begriff von ***Spuren*** einer Aktion, um diese Beobachtung zu formalisieren.

3.4.1 Definition

Spuren bilden die Grundlage für die Lösung der im vorherigen Abschnitt definierten Probleme. Um Rückschlüsse über vergangene Ereignisse in einem digitalen System zu ermöglichen, müssen Spuren jedoch bestimmte Eigenschaften erfüllen, welche wir im Folgenden definieren.

Variablen: {***a***, ***b***}
Initialzustand: {***a*** = 0, ***b*** = 0}
Aktionen:

`p2aktion1`: ***a*** = 0 → ***a*** := 1
`p2aktion2`: ***a*** = 0 → ***a*** := 1
`p2aktion3`: ***a*** = 0 → ***a*** := 1; ***b*** :=1

Abbildung 3.8: ***Programm 2***.

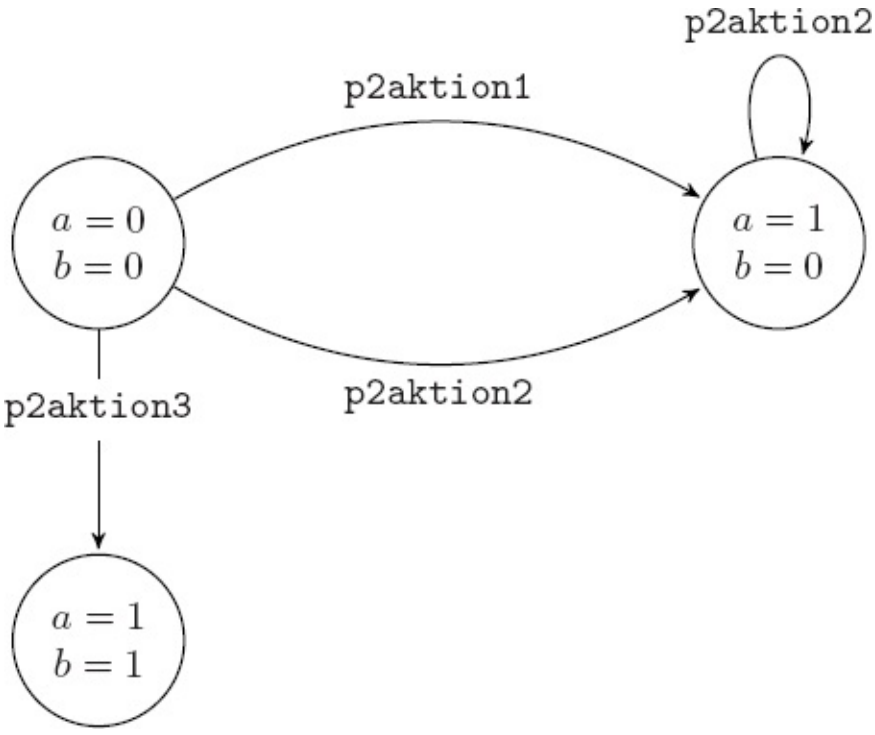


Abbildung 3.9: Zustandsübergangsdiagramm von ***Programm 2***.

Definition 7 (Spuren) ***Sei S*** = (***V***, Σ, q_0) ***ein System und σ*** ∈ Σ ***eine Aktion dieses Systems. Dann ist die*** Spurenmenge (***Evidence Set***, E) ***von σ die Menge***

aller Teilmengen von Wertezuweisungen zu denjenigen Variablen aus V mit den folgenden Eigenschaften:

1. ***Die enthaltenen Zuweisungen werden von σ ausgeführt.***
2. ***Die Menge ist unter Teilmengen abgeschlossen (closed under subsets***).

Wir formulieren die Spurenmenge formal unter Verwendung der Potenzmenge (Menge aller Teilmengen) p bestimmer Variablen-Wert-Paare wie folgt:

$$E(\sigma) = \mathcal{P}(\{[v = d] | [v = d] \in \sigma\}) \tag{3.5}$$

Die Spurenmenge von σ wird verkürzt häufig auch nur als Spuren ***von σ bezeichnet.***

Formal können wir Spuren als Menge boolescher Ausdrücke über dem Zustand des Systems auffassen. Wir sagen, dass eine Spur $\boldsymbol{e} \in \boldsymbol{E}(\boldsymbol{\sigma})$ von Aktion $\boldsymbol{\sigma}$ in Zustand $\boldsymbol{q}$ beobachtet werden kann, wenn der $\boldsymbol{e}$ entsprechende boolesche Ausdruck in $\boldsymbol{q}$ wahr ist.

Wir betrachten nun die verschiedenen Eigenschaften der Spurenmenge im Einzelnen:

Wir fordern in obiger Definition, dass $\boldsymbol{E}(\boldsymbol{\sigma})$ unter Teilmengen abgeschlossen ist. Dies bedeutet, dass alle Teilmengen eines Elementes der Spurenmenge $\boldsymbol{E}(\boldsymbol{\sigma})$ selbst wiederum Elemente dieser Menge sind. Formal ausgedrückt bedeutet dies, dass

$$\forall e \in E(\sigma) : \forall e' \subseteq e : e' \in E(\sigma) \tag{3.6}$$

gilt. Beispielsweise ist die Menge {{$\boldsymbol{a}$ = 1, $\boldsymbol{b}$ = 1}} nicht unter Teilmengen abgeschlossen, und auch die Menge {{$\boldsymbol{a}$ = 1, $\boldsymbol{b}$ = l},{$\boldsymbol{a}$ = 1}} erfüllt dieses Kriterium nicht, da auch alle Teilmengen von {$\boldsymbol{a}$ = 1, $\boldsymbol{b}$ = 1} in $\boldsymbol{E}(\boldsymbol{\sigma})$ enthalten sein müssen. Insbesondere muss also auch die leere Menge Element einer jeden Spurenmenge sein. In diesem Beispiel wäre {{$\boldsymbol{a}$ = 1, $\boldsymbol{b}$ = 1}, {$\boldsymbol{a}$ = 1} , { $\boldsymbol{b}$ = 1}, ∅} eine gültige Spurenmenge. Die Teilmengenabgeschlossenheit erreichen wir durch die Definition über die Potenzmenge.

Da wir zuvor in Abschnitt 3.2.6 voraussetzen, dass eine Aktion zumindest einer Variablen einen Wert zuweisen muss, kann der Fall, dass die Spurenmenge einer Aktion lediglich die leere Menge enthält hier nicht auftreten.

Beispiel 15 (Spuren) ***Beispielsweise ist*** {{***a*** = 1}, ∅} ***eine Spur von*** `p1aktion1` ***in*** Programm 1 ***(siehe Abbildung 3.4). Obgleich man bei Betrachtung des Zustandsübergangsdiagramms in Abbildung 3.6 annehmen könnte, dass*** {b = 0} ***ebenfalls Element der Spurenmenge von*** `p1aktion1` ***ist, nehmen wir zur Kenntnis, dass dies nicht der Fall ist, da*** `p1aktion1` ***keine Zuweisung des Wertes*** 0 ***zur Variablen b vornimmt.***[2] ***Um das Beispiel zu vervollständigen, stellen wir analog fest, dass E***(`p1aktion2`) = {{***b*** = 1}, ∅} ***gilt***.

Variablen: {***a, b***}
Initialzustand: {***a*** = 0, ***b***= 0}
Aktionen:

`p3aktion1`: ***a*** = 0	→	***a*** :=1
`p3aktion2`: ***a*** = 0	→	***a*** :=1; ***b*** := 1

Abbildung 3.10: ***Programm 3***.

Als weiteres Beispiel betrachten wir ***Programm 3*** aus Abbildung 3.10 zusammen mit seiner Darstellung als Zustandsübergangsdiagramm in Abbildung 3.11. Die Spurenmenge der Aktion `p3aktion2` in ***Programm 3*** ist {{***a*** = 1}, {***b*** = 1}, {***a*** = 1, ***b*** = 1}, ∅}. Dies kann als äquivalente Menge boolescher Ausdrücke wie folgt formuliert werden: {{***a*** = 1}, {***b*** = 1}, {***a*** = 1 ∧ ***b*** = 1}, ***TRUE***}. Die leere Menge entspricht hierbei dem booleschen Wert ***TRUE*** und kann offensichtlich in jedem Zustand beobachtet werden. Diese Beobachtung illustriert bereits die mangelnde Aussagekraft von Spuren im Bezug auf vergangene Ereignisse, welche uns unmittelbar zur Definition ***charakteristischer Spuren*** im nächsten Abschnitt führt.

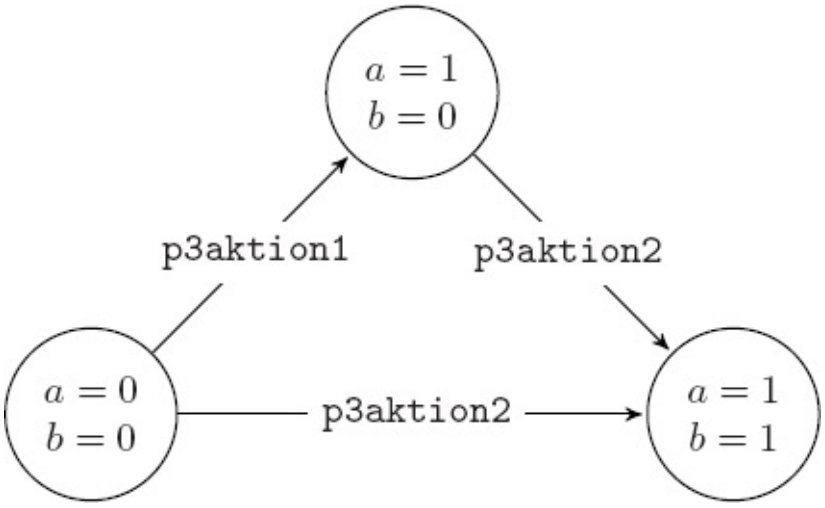


Abbildung 3.11: Zustandsübergangsdiagramm von ***Programm 3***.

3.4.2 Charakteristische Spuren

Wie bereits dargestellt, kann die Spurenmenge einer einzelnen Aktion häufig mehrere Elemente enthalten. Beispielsweise ruft die Aktion p3aktion2 folgende Spuren hervor:

$$E(\texttt{p3aktion2}) = \{\{a = 1, b = 1\}, \{a = 1\}, \{b = 1\}, \varnothing\} \tag{3.7}$$

Weiterhin können unterschiedliche Aktionen durchaus gleiche Spuren hinterlassen. Beispielsweise rufen sowohl `p3aktion1` als auch `p3aktion2` als Element der Spurenmenge (wir sagen auch „als Spur“) die Menge {***a*** = 1} hervor. Formal gilt also {***a*** = 1} ∈ ***E***(`p3aktion1`) ∩ ***E***(`p3aktion2`). Solche Spuren sind also in Bezug auf die erzeugende Aktion nicht eindeutig. Daher definieren wir in diesem Abschnitt den Begriff der ***charakteristischen Spuren***.

Zunächst können wir feststellen, dass Spuren, die bereits im Initialzustand eines Systems enthalten sind, keine Rückschlüsse auf die Ausführung der betreffenden Aktion erlauben, da sie bereits vor Ausführung einer ***beliebigen*** Aktion im System vorhanden sind. Deshalb definieren wir zunächst die Menge all der Spuren, die Teil des Initialzustandes sind und daher gesondert behandelt werden müssen:

Definition 8 (Nullspuren) ***Wir definieren*** Nullspuren (***zero evidence***, ZE) ***als die Menge aller Variablen- Wert-Paare des Initialzustands. Formal gilt also:***

$$ZE = \{[v = d] | [v = d] \in q_0\} \tag{3.8}$$

Für die formale Definition charakteristischer Spuren benötigen wir weiterhin die Notation kombinierter Spuren:

Definition 9 (Kombinierte Spuren) ***Wir definieren*** kombinierte Spuren (***Merged Evidence***, ME) ***einer Menge von Aktionen*** Σ ***als die Vereinigung aller Spuren aller Aktionen in*** Σ.

Formal gilt:

$$ME(\Sigma) = \bigcup_{\sigma \in \Sigma} \bigcup_{e \in E(\sigma)} e \tag{3.9}$$

Definition 10 (Charakteristische Spuren) ***Wir definieren*** charakteristische Spuren (***Characteristic Evidence***, CE) ***einer Aktion*** σ ***bezüglich einer Menge anderer Aktionen*** Σ' ($\sigma \notin \Sigma'$) ***als die maximale Menge von Spuren von*** σ, ***die nicht auch von einer Aktion aus*** Σ' ***hervorgerufen werden oder Teil des Initialzustandes sind***.

Formal gilt also:

$$CE(\sigma, \Sigma') = E(\sigma) \setminus (\mathcal{P}(ME(\Sigma') \cup ZE)) \tag{3.10}$$

Falls σ ebenfalls in Σ' enthalten wäre, so wäre die Menge der charakteristischen Spuren von σ bezüglich Σ' offensichtlich leer. Charakteristische Spuren sind nicht notwendigerweise unter Teilmengen abgeschlossen.

Beispiel 16 (charakteristische Spuren) ***In*** Programm 3 ***hat zum Beispiel die Aktion*** `p3aktion2` ***gegenüber der Aktionsmenge*** {`p3aktion1`} ***die charakteristischen Spuren*** {{***b*** = 1}, {***a*** = 1, ***b*** = 1}}:

$$CE(\texttt{p3aktion2}, \{\texttt{p3aktion1}\}) = E(\texttt{p3aktion2}) \setminus \mathcal{P}(ME(\{\texttt{p3aktion1}\}) \cup ZE) \tag{3.11}$$

Berechnen wir zunächst die Menge der kombinierten Aktionen von {`p3aktion1`}:

$$ME(\{\texttt{p3aktion1}\}) = \bigcup_{e \in \{\{a=1\},\varnothing\}} e \tag{3.12}$$

$$= \{a = 1\} \tag{3.13}$$

Und weiterhin berechnen wir die Menge der Nullspuren in diesem Beispiel:

$$ZE = \{a = 0, b = 0\} \tag{3.14}$$

Damit folgt für die Menge der charakteristischen Spuren:

$$CE(\texttt{p3aktion2}, \{\texttt{p3aktion1}\}) = E(\texttt{p3aktion2}) \setminus \mathcal{P}(ME(\{\texttt{p3aktion1}\}) \cup ZE) \tag{3.15}$$

$$= \{\{a = 1\}, \{b = 1\}, \{a = 1, b = 1\}, \varnothing\} \setminus \mathcal{P}(\{a = 1, a = 0, b = 0\}) \tag{3.16}$$

$$= \{\{a = 1\}, \{b = 1\}, \{a = 1, b = 1\}, \varnothing\} \setminus \{\{a = 1, a = 0, b = 0\}, \{a = 1, a = 0\}, \{a = 1, b = 0\}, \{a = 0, b = 0\}, \{a = 1\}, \{a = 0\}, \{b = 0\}, \varnothing\} \tag{3.17}$$

$$= \{\{b = 1\}, \{a = 1, b = 1\}\} \tag{3.18}$$

In diesem Beispiel stellen die Nullspuren offensichtlich keine Einschränkung der Menge der charakteristischen Spuren dar, da keine Aktionen einer Variable einen Initialwert zuweist. Gleiches gilt für die meisten der nachfolgenden Beispiele, weshalb wir explizit darauf hinweisen, wenn in einem Beispiel eine Initialwertzuweisung stattfindet.

Analog hierzu lässt sich die Menge der charakteristischen Spuren von `p3aktion1` berechnen:

$$CE(\texttt{p3aktion1}, \{\texttt{p3aktion2}\}) = E(\texttt{p3aktion1}) \setminus \mathcal{P}(ME(\{\texttt{p3aktion2}\}) \cup ZE) \tag{3.19}$$

$$= \{\{a = 1\}, \varnothing\} \setminus \mathcal{P}(\{a = 1, b = 1, a = 0, b = 0\}) \tag{3.20}$$

$$= \varnothing \tag{3.21}$$

Die Aktion `p3aktion1` besitzt also keine charakteristischen Spuren gegenüber der Aktion `p3aktion2`.

Aus der Definition der charakteristischen Spuren folgt außerdem die folgende Beobachtung:

$$(CE(\sigma, \Sigma') \neq \varnothing) \ \wedge \ (\Sigma'' \subseteq \Sigma') \quad \Rightarrow \quad CE(\sigma, \Sigma'') \neq \varnothing \tag{3.22}$$

Dies beschreibt eine Monotonie-Eigenschaft charakteristischer Spuren und bedeutet insbesondere, dass charakteristische Spuren umso aussagekräfiger werden, je größer Σ′ ist, da sie damit auch in Bezug auf alle Teilmengen von Σ′ gelten. Sei beispielsweise Σ = {$\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, $\boldsymbol{\sigma}_3$, $\boldsymbol{\sigma}_4$, $\boldsymbol{\sigma}_5$} und $\boldsymbol{e}_1 \in \boldsymbol{CE}(\boldsymbol{\sigma}_1, \{\boldsymbol{\sigma}_2, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \boldsymbol{\sigma}_5\})$. Dann gilt beispielsweise, dass $\boldsymbol{e}_1$ auch charakteristische Spur von $\boldsymbol{\sigma}_1$ bezüglich der Menge {$\boldsymbol{\sigma}$2, $\boldsymbol{\sigma}$3, $\boldsymbol{\sigma}$4} ist: $\boldsymbol{e}_1 \in \boldsymbol{CE}(\boldsymbol{\sigma}_1, \{\boldsymbol{\sigma}_2, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4\})$. Hingegen könnte die Spur $\boldsymbol{e}_2 \in \boldsymbol{CE}(\boldsymbol{\sigma}_1, \{\boldsymbol{\sigma}_2, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4\})$ durch Ausführung der Aktion $\boldsymbol{\sigma}_5$ zunichte gemacht werden, wenn $\boldsymbol{e}_2$ Spur von $\boldsymbol{\sigma}_5$ ist und damit $\boldsymbol{e}_2 \notin \boldsymbol{CE}(\boldsymbol{\sigma}_1, \{\boldsymbol{\sigma}_2, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \boldsymbol{\sigma}_5\})$ gilt.

3.4.3 Gemeinsame charakteristische Spuren

Als nächsten Schritt betrachten wir solche Spuren, die nicht nur für eine einzelne Aktion, sondern für eine Menge von Aktionen charakteristisch sind.

Definition 11 (Gemeinsame charakteristische Spuren) ***Seien S*** = (***V***, Σ, $\boldsymbol{q}_0$) ***ein System und*** Σ′ ⊆ Σ ***und*** Σ″ ⊆ Σ ***zwei Mengen von Aktionen aus S. Wir definieren*** gemeinsame charakteristische Spuren (***Common Characteristic Evidence***, CCE) ***von*** Σ′ ***bezüglich*** Σ″ ***als die Vereinigung der charakteristischen Spuren aller Aktionen*** σ′ ∈ ***Σ′ bezüglich*** Σ″. ***Formal ausgedrückt gilt also:***

$$CCE(\Sigma', \Sigma'') = \bigcup_{\sigma' \in \Sigma'} CE(\sigma', \Sigma'') \tag{3.23}$$

Zur Veranschaufichung gemeinsamer charakteristischer Spuren benötigen wir ein etwas komplexeres Beispiel mit wenigstens drei unterschiedlichen Aktionen. Daher betrachten wir nun ***Programm 4*** aus Abbildung 3.12. Das zugehörige Diagramm ist in Abbildung 3.13 gegeben.

Variablen: {***a, b, c***}
Initialzustand: {***a*** = 0, ***b*** = 0, ***c*** = 0}
Aktionen:

`p4aktion1`: ***a*** = 0 → ***a*** := 1; ***c*** := 1
`p4aktion2`: ***b*** = 0 → ***a*** := l; ***b*** := 1
`p4aktion3`: ***b*** = 1 ∧ c = 0 → ***b*** := l; ***c*** := l

Abbildung 3.12: ***Programm 4.***

Beispiel 17 (Gemeinsame charakteristische Spuren) ***Beispielsweise haben die beiden Aktionen `p4aktion1` und `p4aktion2` gegenüber `p4aktion3` die gemeinsame charakteristische Spur*** {{***a*** = 1}}:

$$\begin{aligned} &CCE(\{\texttt{p4aktion1}, \texttt{p4aktion2}\}, \{\texttt{p4aktion3}\}) \\ &\quad = CE(\texttt{p4aktion1}, \{\texttt{p4aktion3}\}) \cup CE(\texttt{p4aktion2}, \{\texttt{p4aktion3}\}) \end{aligned} \tag{3.24}$$

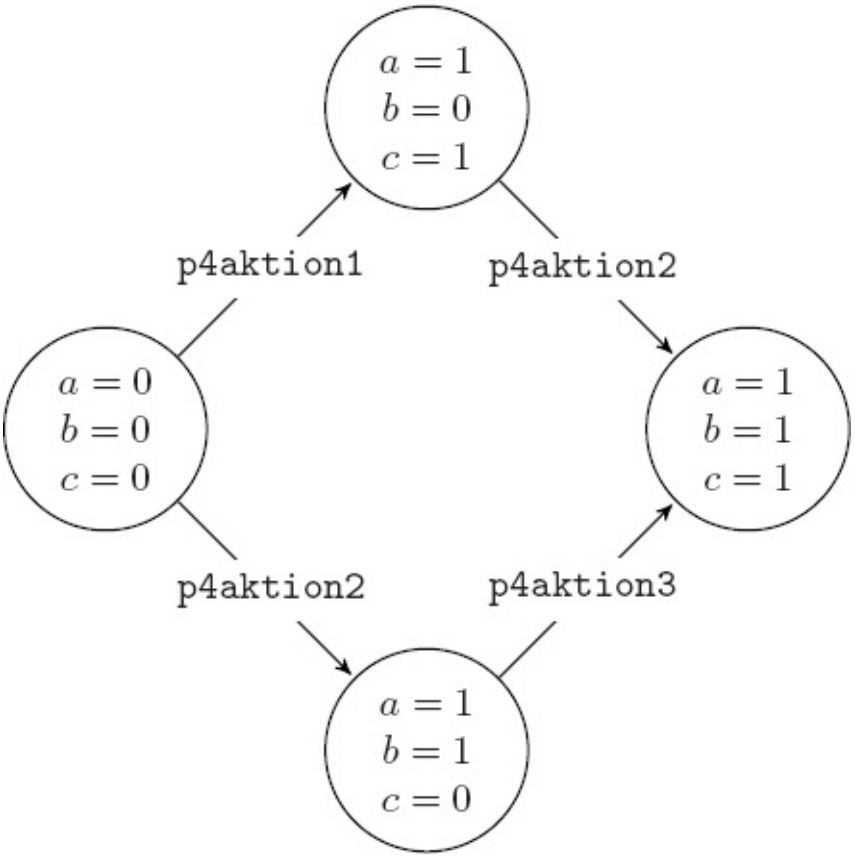


Abbildung 3.13: Zustandsübergangsdiagramm von ***Programm 4***.

Wir berechnen also zunächst die charakteristischen Spuren der Aktionen `p4aktion1` und `p4aktion2` – jeweils gegenüber {`p4aktion3`}:

$$CE(\texttt{p4aktion1}, \{\texttt{p4aktion3}\}) = E(\texttt{p4aktion1}) \setminus \mathcal{P}(ME(\{\texttt{p4aktion3}\}) \cup ZE) \tag{3.25}$$

$$= \{\{a = 1\}, \{c = 1\}, \{a = 1, c = 1\}, \varnothing\} \setminus \mathcal{P}(\{b = 1, c = 1\} \cup \{a = 0, b = 0, c = 0\} \tag{3.26}$$

$$= \{\{a = 1\}, \{a = 1, c = 1\}\} \tag{3.27}$$

Die charakteristischen Spuren von Aktion `p4aktion2` gegenüber `p4aktion3` sind dann:

$$CE(\texttt{p4aktion2}, \{\texttt{p4aktion3}\}) = E(\texttt{p4aktion2}) \setminus \mathcal{P}(ME(\{\texttt{p4aktion3}\}) \cup ZE) \tag{3.28}$$

$$= \{\{a=1\},\{b=1\},\{a=1,b=1\},\varnothing\} \setminus \mathcal{P}(\{b=1,c=1\} \cup \{a=0,b=0,c=0\} \tag{3.29}$$

$$= \{\{a=1\},\{a=1,b=1\}\} \tag{3.30}$$

Mit den Ergebnissen der Gleichungen 3.25 bis 3.27 und Gleichungen 3.28 bis 3.30 folgt:

$$\begin{aligned} &CCE(\{p4aktion1, p4aktion2\}, \{p4aktion3\}) \\ &\quad = CE(p4aktion1, \{p4aktion3\}) \cup CE(p4aktion2, \{p4aktion3\}) \\ &\quad = \{\{a=1\},\{a=1,c=1\}\} \cup \{\{a=1\},\{a=1,b=1\}\} \\ &\quad = \{\{a=1\},\{a=1,c=1\},\{a=1,b=1\}\} \end{aligned} \tag{3.31}$$

3.5 Kontraspuren

Wie wir in den folgenden Abschnitten im Detail erläutern werden, können bei der Rekonstruktion von Ereignissen in einem digitalen System nicht nur Spuren von Relevanz sein, sondern auch sogenannte Kontraspuren, die es im Wesentlichen erlauben, die Ausführung bestimmter Aktionen auszuschließen.

3.5.1 Definition

Am Beispiel von Aktion `p1aktion1` in ***Programm 1*** (siehe Abbildung 3.4) konnten wir bereits intuitiv beobachten, dass der Inhalt von Variablen auch dann Schlussfolgerungen ermöglichen kann, wenn diese nicht Spur einer Aktion sind. In besagtem Beispiel ist es offensichtlich, dass wann immer in einem Zustand ***b*** = 0 gilt, ausgeschlossen werden kann, dass Aktion `p1aktion2` ausgeführt wurde. Diese Erkenntnis erlaubt nach dem Ausschlussprinzip wiederum weitere Schlussfolgerungen, nämlich dass in einem solchen Zustand entweder überhaupt keine Aktion ausgeführt wurde, oder aber `p1aktion1`. Die Beobachtung, dass ***b*** = 0 gilt, birgt also ebenfalls wichtige Informationen zur Rekonstruktion der Ereignis-Historie des Systems. Es handelt sich hier in der Tat um eine weitere Art von Spuren, welche dazu geeignet ist, die Ausführung einer Aktion zu ***widerlegen***. Wir formalisieren diese Art von Spuren durch den Begriff der ***Kontraspuren***.

Definition 12 (Kontraspuren) ***Sei S*** = (***V***, Σ, q_0) ***ein System und σ*** ∈ Σ ***eine Aktion dieses Systems. Dann ist die*** Kontraspurenmenge ***(Counter Evidence***, XE***) von σ diejenige Menge von Teilmengen von Variablen aus V zusammen mit einer Zuweisung von Werten zu diesen Variablen mit den folgenden Eigenschaften:***

1. ***Die Aktion σ weist den enthaltenen Variablen einen Wert zu***.
2. ***Die Zuweisungen von Werten werden von σ nicht ausgeführt***.
3. ***Die Menge ist unter Teilmengen abgeschlossen***.

Formal gilt also:

$$XE(\sigma) = \mathcal{P}\Big(\bigcup_{[v=d]\mid\exists d' \neq d:[v=d']\in\sigma} \{[v=d]\}\Big) \qquad (3.32)$$

Die Elemente der Kontraspurenmenge werden auch als Kontraspuren ***von σ bezeichnet***.

Es handelt sich bei den Kontraspuren einer Aktion also um eine Menge von Teilmengen von Variablen aus ***V*** , denen ***σ*** Werte zuweist, zusammen mit der Zuweisung all derjenigen Werte aus der Domäne dieser Variablen, die diesen Variablen von ***σ nicht*** zugewiesen werden.

Kontraspuren stellen eine Art „Negation“ der Spuren dar und beschreiben Veränderungen am Zustand des Systems die von ***σ nicht*** ausgeführt werden, während Spuren gerade solche Änderungen des Zustandes enthalten, die durch die Ausführung von ***σ*** hervorgerufen werden.

Beispiel 18 (Kontraspuren) ***Beispielsweise ist*** {{***a*** = 0}, ∅} ***Kontraspurenmenge von Aktion*** `p1aktion1` ***in*** Programm 1 (***siehe Abbildung 3.4 auf Seite →), da*** `p1aktion1` ***der Variablen a einen Wert zuweist (nämlich*** 1***). Da die Domäne D*** = {0,1} ***ist, verbleibt die Zuweisung des Wertes*** 0 ***zur Variablen a als Kontraspur. Da*** `p1aktion1` ***keiner weiteren Variablen einen Wert zuweist, wird auch für die Kontraspurenmenge keine weitere Variable berücksichtigt. Weiterhin muss die Kontraspurenmenge unter Teilmengen abgeschlossen sein, so dass in diesem Beispiel die leere Menge als Element der Kontraspurenmenge hinzukommt. Analog hierzu gilt für*** Programm 1:

$$XE(\texttt{p1aktion2}) = \{\{b = 0\}, \varnothing\} \tag{3.33}$$

Beispiel 19 (Kontraspuren) ***Als etwas komplexeres Beispiel von Kontraspuren betrachten wir*** Programm3. ***Hier ist*** {{***a*** = 0}, ∅} ***Kontraspur von Aktion*** `p3aktion1`***. Weiterhin sind die Kontraspuren von*** `p3aktion2` ***durch die Menge*** {{***a*** = 0}, {***b*** = 0}, {***a*** = 0, ***b*** = 0}, ∅} ***gegeben. Das bedeutet in diesem Beispiel, dass wenn in einem Zustand b*** = 0 ***gilt, die Aktion*** `p3aktion2` ***nicht ausgeführt worden sein kann. Diese Beobachtung gilt jedoch einzig, weil es im betrachteten Beispiel keine Aktion gibt, welche die Zuweisung b*** = 0 ***ausführt, denn dann könnte die Variable b nach Ausführung der Aktion*** `p3aktion2` ***wieder auf den Wert*** 0 ***zurückgesetzt worden sein. In allen bisher eingeführten Beispielen weisen Aktionen den Variablen jedoch lediglich den Wert*** 1 ***zu, und der Initialwert ist stets*** 0. ***In Beispielen mit Zuweisungen des Initialwertes oder einer Domäne mit mehr als zwei unterschiedlichen Werten stoßen wir auf das angesprochene Problem, dass zur Kontraspur einer Aktion gehörige Werte auch von anderen Aktionen erzeugt werden könnten. Daher ist die Definition von Kontraspuren nicht ausreichend, um die Ausführung bestimmter Aktionen generell widerlegen zu können. Wir demonstrieren dies am Beispiel von*** Programm 5 ***aus Abbildung 3.14. Zu beachten ist in diesem Beispiel besonders, dass*** `p5aktion2` ***der Variablen a den Wert*** 0 ***zuweist. Zur Veranschaufichung zeigt Abbildung 3.15 das zugehörige Zustandsübergangsdiagramm***.

Variablen: {***a, b***}
Initialzustand: {***a*** = 0, ***b*** = 0}
Aktionen:

`p5aktion1`: ***a*** = 0 → ***a*** := 1
`p5aktion2`: ***b*** = 0 → ***a*** := 0; ***b*** := 1

Abbildung 3.14: ***Programm 5***.

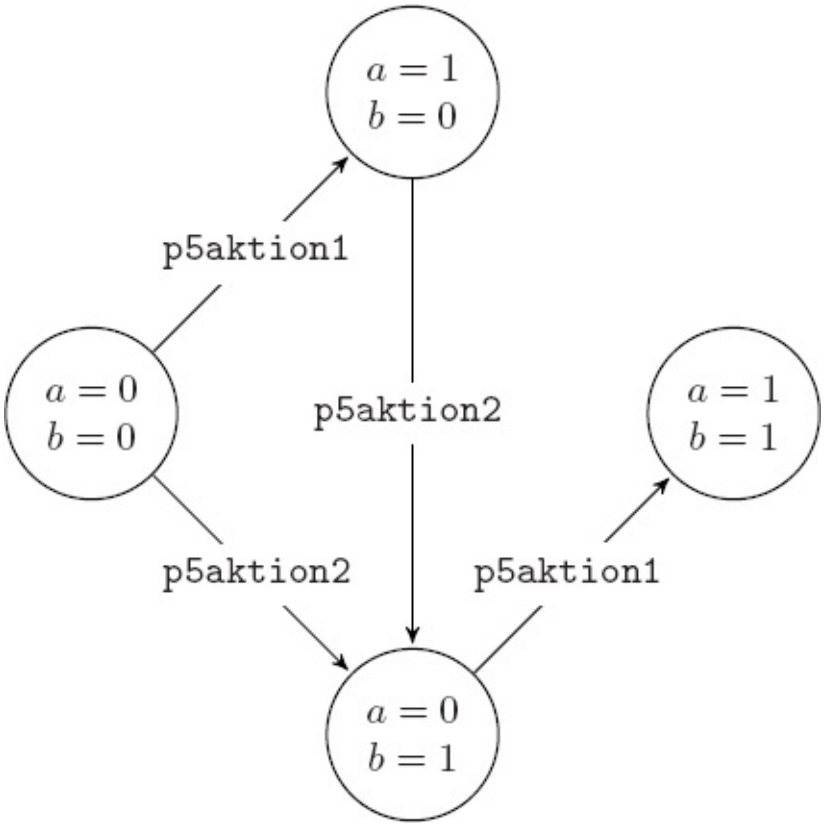


Abbildung 3.15: Zustandsübergangsdiagramm von ***Programm 5***.

Aktion `p5aktion1` hat hier die Kontraspur {{***a*** = O}, ∅}. Daher könnte man geneigt sein, im Zustand {***a*** = 0, ***b*** = 1} die vorherige Ausführung der Aktion `p5aktion1` auszuschließen. Wie jedoch Abbildung 3.15 zeigt, kann dieser Zustand ebenso nach Ausführung der Aktionen `p5aktion1` und `p5aktion2` erreicht werden. Um dieser Problematik zu begegnen, sind analog zu ***charakteristischen Spuren*** auch ***charakteristische Kontraspuren*** erforderlich, welche wir im nächsten Abschnitt einführen.

3.5.2 Charakteristische Kontraspuren

Wie im vorherigen Abschnitt diskutiert, sind ***Kontraspuren*** allein nicht ausreichend, um die Ausführung einer konkreten Aktion generell ausschließen zu können. Vielmehr erfordert dies zusätzliche Bedingungen, welche Kontraspuren als aussagekräftigere ***charakteristische Kontraspuren*** auszeichnen. Diese werden im Folgenden definiert.

Definition 13 (Charakteristische Kontraspuren) ***Wir definieren*** charakteristische Kontraspuren (***Characteristic Counter Evidence***, CXE***) als diejenigen Kontraspuren, die nicht auch durch die Ausführung einer anderen Aktion entstehen können. Formal gilt also:***

$$\boldsymbol{CXE}(\boldsymbol{\sigma}, \Sigma') = \boldsymbol{XE}(\boldsymbol{\sigma}) \setminus \boldsymbol{P}(\boldsymbol{ME}(\Sigma'))$$

Beispiel 20 (charakteristische Kontraspuren) ***Vollziehen wir dies nun am Beispiel von*** Programm 5 ***nach, so sehen wir, dass Aktion*** `p5aktion1` ***bezüglich*** `p5aktion2` ***keine charakteristischen Kontraspuren besitzt:***

$$CXE(\texttt{p5aktion1}, \{\texttt{p5aktion2}\}) = XE(\texttt{p5aktion1}) \setminus \mathcal{P}(\ ME(\{\texttt{p5aktion2}\})\ ) \tag{3.34}$$

$$= \{\{a = 0\}, \varnothing\} \setminus \mathcal{P}(\{a = 0, b = 1\}) \tag{3.35}$$

$$= \{\{a = 0\}, \varnothing\} \setminus \{\{a = 0, b = 1\}, \{a = 0\}, \{b = 1\}, \varnothing\} \tag{3.36}$$

$$= \varnothing \tag{3.37}$$

Hingegen hat Aktion `p5aktion2` ***bezüglich*** `p5aktion1` ***die charakteristische Kontraspur*** $\{\{\boldsymbol{b} = 0\}, \{\boldsymbol{a} = 1, \boldsymbol{b} = 0\}\}$:

$$CXE(\texttt{p5aktion2}, \{\texttt{p5aktion1}\}) = XE(\texttt{p5aktion2}) \setminus \mathcal{P}(ME(\{\texttt{p5aktion1}\})) \tag{3.38}$$

$$= \{\{a = 1\}, \{b = 0\}, \{a = 1, b = 0\}, \varnothing\} \setminus \mathcal{P}(\{a = 1\}) \tag{3.39}$$

$$= \{\{a = 1\}, \{b = 0\}, \{a = 1, b = 0\}, \varnothing\} \setminus \{\{a = 1\}, \varnothing\} \tag{3.40}$$

$$= \{\{b = 0\}, \{a = 1, b = 0\}\} \tag{3.41}$$

Existieren, wie in den zuvor angeführten Beispielen, keine Zuweisungen des Initialwertes oder eine mehrwertige Domäne, so entsprechen die ***charakteristischen Kontraspuren*** den ***Kontraspuren*** ohne die leere Menge. Dies lässt sich an ***Programm 1*** demonstrieren, da hier ***XE***(`p1aktion1`) ∩ ***E***(`p1aktion2`) = {∅} gilt:

$$CXE(\texttt{p1aktion1}, \{\texttt{p1aktion2}\}) = XE(\texttt{p1aktion1}) \setminus \mathcal{P}(\ ME(\{\texttt{p1aktion2}\})\ ) \quad (3.42)$$

$$= \{\{a = 0\}, \varnothing\} \setminus \{\{b = 1\}, \varnothing\} \quad (3.43)$$

$$= \{\{a = 0\}\} \quad (3.44)$$

$$= XE(\texttt{p1aktion1}) \setminus \{\varnothing\} \quad (3.45)$$

3.5.3 Gemeinsame charakteristische Kontraspuren

Schließlich widmen wir uns nun ***gemeinsamen charakteristischen Kontraspuren*** einer Menge von Aktionen.

Definition 14 (Gemeinsame charakteristische Kontraspuren) Gemeinsame charakteristische Kontraspuren ***(Common Characteristic Counter Evidence***, CCXE***) einer Menge von Aktionen sind diejenigen charakteristischen Kontraspuren, die alle Aktionen dieser Menge gemeinsam haben. Formal gilt:***

$$CCXE(\Sigma', \Sigma'') = \bigcap_{\sigma' \in \Sigma'} CXE(\sigma', \Sigma'') \quad (3.46)$$

Beispiel 21 (gemeinsame charakteristische Kontraspuren) ***Als Beispiel für gemeinsame charakteristische Kontraspuren betrachten wir das*** Programm 4 aus ***Abbildung 3.12. Hier haben beispielsweise die beiden Aktion `p4aktion1` und `p4aktion2` gegenüber `p4aktion3` die gemeinsame charakteristische Kontraspur*** {{***a=0***}}:

$$CCXE(\{p4aktion1, p4aktion2\}, \{p4aktion3\})$$
$$= CXE(p4aktion1, \{p4aktion3\}) \cap CXE(p4aktion2, \{p4aktion3\}) \quad (3.47)$$

Wir berechnen zunächst die charakteristischen Kontraspuren von `p4aktion1` und `p4aktion2` jeweils gegenüber {***`p4aktion3`***}:

$$CXE(\texttt{p4aktion1}, \{\texttt{p4aktion3}\}) = XE(\texttt{p4aktion1}) \setminus \mathcal{P}(\ ME(\{\texttt{p4aktion3}\})\ ) \quad (3.48)$$

$$= \{\{a=0\},\{c=0\},\{a=0,c=0\},\varnothing\} \setminus \{\{b=1\},\{c=1\},\{b=1,c=1\},\varnothing\} \quad (3.49)$$

$$= \{\{a=0\},\{c=0\},\{a=0,c=0\}\} \quad (3.50)$$

Die charakteristischen Kontraspuren von `p4aktion2` gegenüber {`p4aktion3`} lassen sich dann analog berechnen:

$$CXE(\texttt{p4aktion2}, \{\texttt{p4aktion3}\}) = XE(\texttt{p4aktion2}) \setminus \mathcal{P}(\ ME(\{\texttt{p4aktion3}\})\ ) \quad (3.51)$$

$$= \{\{a=0\},\{b=0\},\{a=0,b=0\},\varnothing\} \setminus \{\{b=1\},\{c=1\},\{b=1,c=1\},\varnothing\} \quad (3.52)$$

$$= \{\{a=0\},\{b=0\},\{a=0,b=0\}\} \quad (3.53)$$

Mit den Ergebnissen der Gleichungen 3.48 bis 3.50 und Gleichungen 3.51 bis 3.53 folgt:

$$\begin{aligned} CCXE(\{\texttt{p4aktion1}, \texttt{p4aktion2}\}, \{\texttt{p4aktion3}\}) &= CXE(\texttt{p4aktion1}, \{\texttt{p4aktion3}\}) \cap CXE(\texttt{p4aktion2}, \{\texttt{p4aktion3}\}) \\ &= \{\{a=0\},\{c=0\},\{a=0,c=0\}\} \cap \{\{a=0\},\{b=0\},\{a=0,b=0\}\} \\ &= \{\{a=0\}\} \end{aligned} \quad (3.54)$$

Bei gemeinsamen charakteristischen Kontraspuren handelt es sich um diejenigen Spuren, die nicht nur die Ausführung einer Aktion widerlegen, sondern einer Menge von Aktionen. Zunächst hat diese Art von Spuren scheinbar keinen erkennbar größeren Nutzen als ***charakteristische Kontraspuren***, da in der Praxis üblicherweise Spuren einer Aktion gefunden werden und daraufhin alternative Hypothesen (andere Aktionen, die diese Spuren erzeugt haben könnten) gezielt durch das Auffinden von charakteristischen Kontraspuren ausgeschlossen werden (vgl. Kapitel 1, Die wissenschaftliche Methode). Falls derartige Kontraspuren existieren, ist es in der Regel nicht von Belang, ob diese gleichzeitig dazu geeignet wären, weitere

Aktionen ausschließen. Betrachten wir jedoch den Fall, in dem verschiedene Aktionen gleiche Kontraspuren besitzen und daher einander gegenüber keine ***charakteristischen*** Kontraspuren besitzen. Auch wenn sich derartige Aktionen hinsichtlich ihrer Kontraspuren nicht unterscheiden lassen, so verbleibt die Option, die Menge dieser Aktionen als Einheit zu betrachten, welche dann durchaus gemeinsame charakteristische Kontraspuren erzeugen kann und so von anderen Aktionen unterschieden werden kann. Auf die genaue Bedeutung von Spuren und Kontraspuren für die Rekonstruierbarkeit vergangener Ereignisse in einem digitalen System gehen wir jedoch erst in Abschnitt 3.7 im Detail ein und widmen uns nun der Frage, wann diese für die Rekonstruktion wichtigen Spuren entstehen.

3.6 Interferenz

Anhand einiger Beispielprogramme der vorangegangenen Abschnitte konnten wir bereits feststellen, dass es unter bestimmten Umständen möglich ist, Pfade zu rekonstruieren und damit Aussagen über frühere Zustände und insbesondere stattgefundene Aktionen zu machen. Wir untersuchen nun Voraussetzungen für derartige Aussagen.

Eine Eigenschaft von ***Programm 1*** aus Abbildung 3.4 und ein möglicher Grund für die vollständige Rekonstruierbarkeit der gesamten System-Historie in jedem möglichen Zustand ist die Tatsache, dass die beiden Aktionen `p1aktion1` und `p1aktion2` auf disjunkten Mengen von Variablen operieren. Wir zeigen nun, dass dies jedoch keine notwendige Voraussetzung für die Rekonstruierbarkeit vorausgegangener Aktionen ist.

Variablen: {***a, b, c***}
Initialzustand: {***a*** = 0, ***b*** = 0, ***c*** = 0}
Aktionen:

`p6aktion1`: ***a*** = 0 → ***a*** := 1; ***b*** := 1
`p6aktion2`: ***c*** = 0 → ***b*** := 1; ***c*** := 1

Abbildung 3.16: ***Programm 6***.

Betrachten wir ***Programm 6*** aus Abbildung 3.16. Genau wie ***Programm 1*** verwendet ***Programm 6*** zwei Aktionen. Anders als ***Programm 1*** besitzt

Programm 6 jedoch drei Variablen, und die Kommandos der beiden Aktionen operieren beide auf der Variablen ***b***. Beide Aktionen weisen also derselben Variablen einen Wert zu. Trotzdem fällt bei Betrachtung der beiden zugehörigen Zustandsübergangsdiagramme in den Abbildungen 3.6 und 3.17 auf, dass ***Programm 6*** im Wesentlichen das gleiche Verhalten zeigt wie ***Programm 1***. Auch die Möglichkeiten der Rekonstruktion sind identisch: In jedem Zustand können jeweils die gleichen Aussagen über die vorherige Ausführung der beiden Aktionen `p1aktion1` und `p1aktion2` beziehungsweise `p6aktion1` und `p6aktion2` getroffen werden:

- In Zustand {***a*** = 0, ***b*** = 0, ***c*** = 0} wissen wir, dass keine der beiden Aktionen stattgefunden hat.
- In Zustand {***a*** = 1, ***b*** = 1, ***c*** = 0} wissen wir, dass Aktion `p6aktion1` und nur diese Aktion stattgefunden hat.
- In Zustand {***a*** = 0, ***b*** = 1, ***c*** = 1} wissen wir, dass Aktion `p6aktion2` und nur diese Aktion stattgefunden hat.
- In Zustand {***a*** = 1, ***b*** = 1, ***c*** = 1} wissen wir, dass beide Aktionen stattgefunden haben, jedoch ist die Reihenfolge ihrer Ausführung nicht rekonstruierbar.

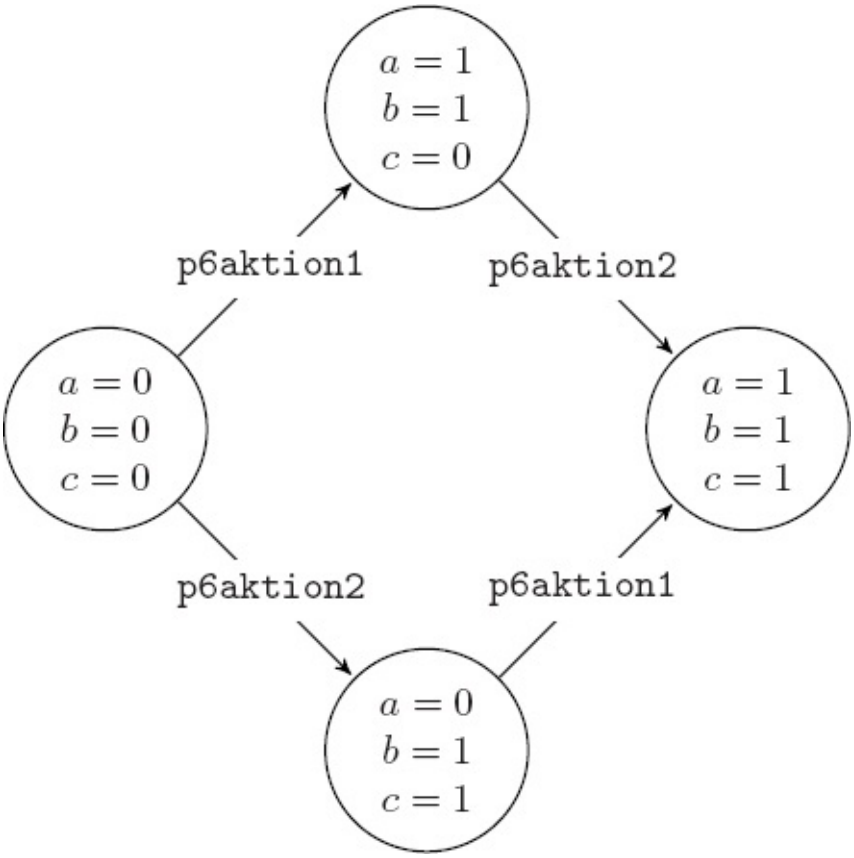


Abbildung 3.17: Zustandsübergangsdiagramm von ***Programm 6***.

Es ist also keine notwendige Voraussetzung für die Rekonstruktion der ausgeführten Aktionen, dass die Mengen der Variablen, auf denen Aktionen operieren, disjunkt sind. Vielmehr scheint es relevant zu sein, dass diese Mengen sich nicht vollständig überlappen. Wir nennen eine solche Überlappung ***Interferenz*** (***Interference***).

Zunächst führen wir diese Interferenz allgemein auf einer beliebigen endlichen Menge ein und werden sie später zwei mal instanziieren: Einerseits wie hier motiviert auf Mengen von Variablen und andererseits als Verfeinerung auf Mengen von Spuren.

3.6.1 Allgemeine Interferenz

Wir unterscheiden die folgenden drei Grade von Interferenz zunehmender Stärke:

- Schwache Interferenz

- Starke Interferenz
- Absolute Interferenz

Diese Grade der Interferenz werden in den folgenden Abschnitten in der Reihenfolge zunehmender Stärke der Überlappung erläutert. Der Fall der fehlenden Interferenz kann an sich nicht als Grad der Interferenz bezeichnet werden. Dennoch möchten wir auch diesen Fall nicht außer Acht lassen und werden ihn unter dem Namen der ***Nicht-Interferenz*** erörtern.

Striktheit

Wir definieren Interferenz im Folgenden in stärker werdenden Stufen. Tatsächlich wird eine Stufe der Interferenz immer von der nächst stärkeren Stufe impliziert. Daher bezeichnen wir die Existenz einer bestimmten Stufe der Interferenz ohne die Existenz einer stärkeren Form der Interferenz als ***strikt***. Beispielsweise bezeichnet ***strikt schwache Interferenz*** das Auftreten schwacher Interferenz, jedoch keiner starken Interferenz.

Nicht-Interferenz

Die Nicht-Interferenz zweier endlicher Mengen ist dann gegeben, wenn sie keine gemeinsamen Elemente enthalten.

Definition 15 (Nicht-Interferenz) ***Seien M und N zwei endliche Mengen. Es herrscht*** Nicht-Interferenz (***Non-Interference) zwischen den Mengen M und N, wenn die Schnittmenge von M und N leer ist, also wenn M und N disjunkt sind:***

$$M \cap N = \varnothing \tag{3.55}$$

In diesem Fall bezeichnen wir das System als interferenzfrei.

Abbildung 3.18 auf der nächsten Seite illustriert zur Veranschaufichung schematisch zwei sich nicht überschneidende Mengen ***M*** und ***N***. Der Übersichtlichkeit wegen werden in Abbildung 3.23 neben der aus Euler-Diagrammen bekannten zweidimensionalen Darstellung der Mengen als Ellipsen auch die einzelnen Mengen eindimensional nebeneinander dargestellt.

Insbesondere starke Formen der Interferenz lassen sich auf diese Weise deutlicher darstellen.

Schwache Interferenz

Das Vorhandensein schwacher Interferenz zwischen zwei Mengen bedeutet intuitiv, dass es im Gegensatz zur Nicht-Interferenz Elemente gibt, welche in beiden Mengen enthalten sind. Dies stellt die Bedingung für schwache Interferenz als schwächste Form der Interferenz dar.

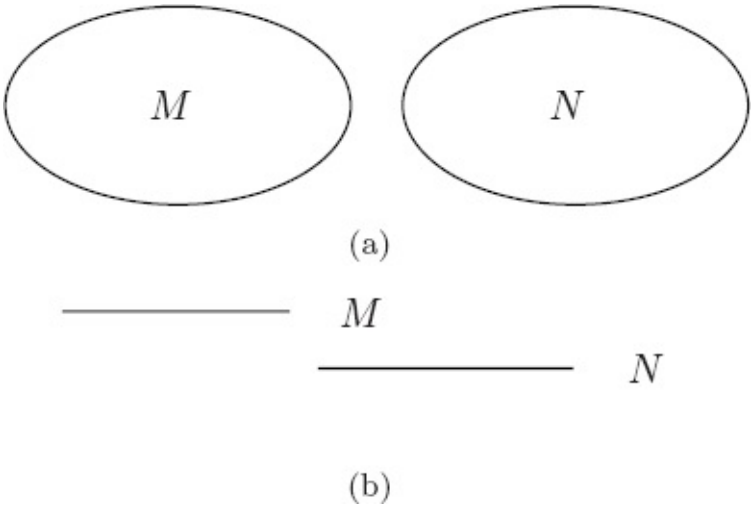


Abbildung 3.18: Schematische Darstellung zweier Mengen bei ***Nicht-Interferenz***.

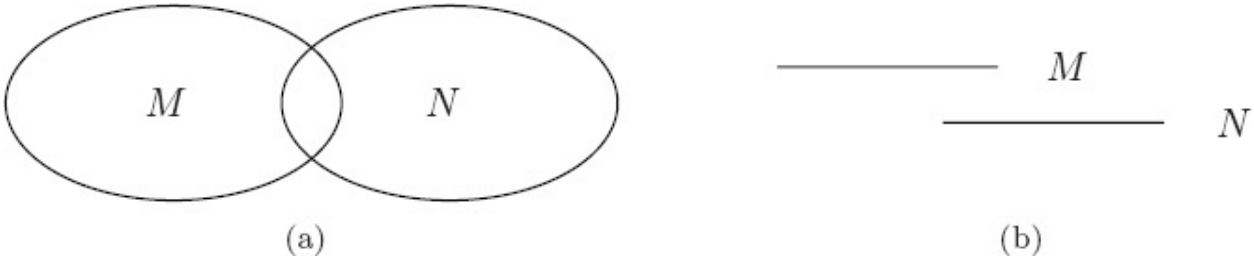


Abbildung 3.19: Schematische Darstellung zweier Mengen ***M*** und ***N*** bei ***schwacher Interferenz***.

Definition 16 (Schwache Interferenz) ***Seien M und N zwei endliche Mengen. Es herrscht*** schwache Interferenz (***Weak Interference) zwischen den Mengen M und N , wenn die Schnittmenge von M und N nicht leer ist:***

$$M \cap N \neq \varnothing \tag{3.56}$$

Abbildung 3.19 zeigt die teilweise Überlappung der Mengen ***M*** und ***N*** .

Starke Interferenz

Die nächst stärkere Form der Interferenz zweier Mengen stellt die ***starke Interferenz*** dar, bei der eine Menge vollständig in der anderen Menge enthalten ist, wie in Abbildung 3.20 auf der nächsten Seite schematisch dargestellt.

Definition 17 (Starke Interferenz) ***Seien M und N zwei endliche Mengen. Es herrscht*** starke Interferenz (***Strong Interference) zwischen den Mengen M und N, wenn die Schnittmenge von M und N eine Obermenge von M und M damit vollständig in N enthalten ist:***

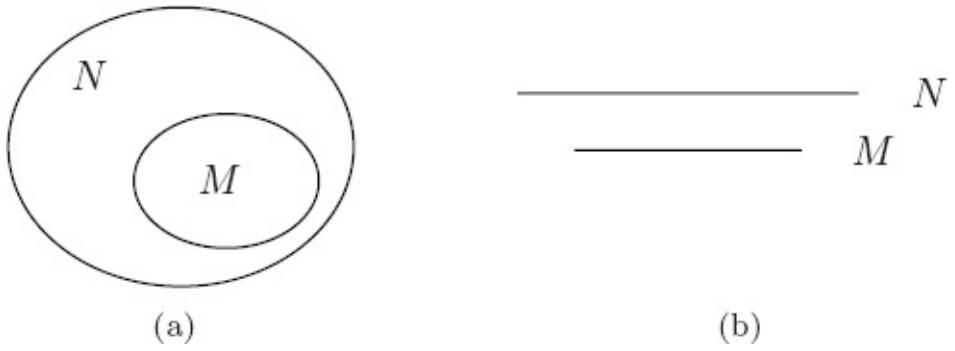


Abbildung 3.20: Schematische Darstellung zweier Mengen mit ***starker Interferenz***.

$$M \cap N \supseteq M \Leftrightarrow M \subseteq N \tag{3.57}$$

Absolute Interferenz

Absolute Interferenz stellt die stärkste Form der Interferenz zweier Mengen dar, bei der sich beide Mengen gegenseitig vollständig enthalten, also beide Mengen gleich sind.

Definition 18 (Absolute Interferenz) ***Seien M und N zwei endliche Mengen. Es herrscht*** absolute Interferenz (***Total Interference) zwischen den Mengen M und N , wenn die Schnittmenge von M und N sowohl eine Obermenge von M, als auch von N ist und M und N damit gleich sind:***

$$(M \cap N \supseteq M) \wedge (M \cap N \supseteq N) \Leftrightarrow M = N \tag{3.58}$$

Abbildung 3.21 zeigt schematisch die vollständige wechselseitige Überlappung zweier Mengen ***M*** und ***N*** .

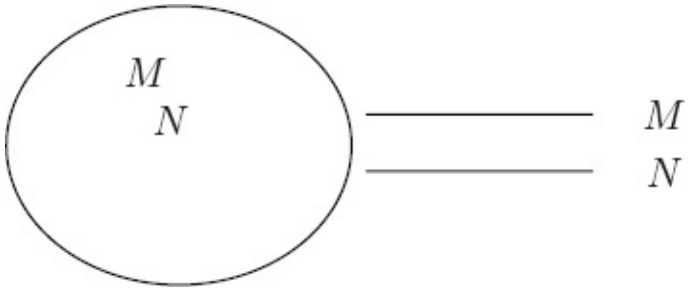


Abbildung 3.21: Schematische Darstellung zweier Mengen ***M*** und ***N*** bei ***absoluter Interferenz***.

Im Folgenden werden wir nun diese allgemeinen Grade der Interferenz zugrunde legen, um Interferenz von Aktionen in Bezug auf genutzte Variablen und in Bezug auf die erzeugten Spuren zu betrachten.

3.6.2 Variablen-Interferenz

Bei der Variablen-Interferenz geht es um die Interferenz zwischen Mengen von Variablen: Sei Σ die Menge aller Aktionen eines Systems ***S*** wie zuvor und seien $\sigma \in \Sigma$ eine Aktion dieses Systems und $\Sigma' \subset \Sigma$ eine Menge anderer Aktionen ($\sigma \notin \Sigma'$). Dann setzen wir ***M*** gleich der Menge der von Aktion ***σ*** veränderten Variablen:

$$M = cvars(\sigma) \tag{3.59}$$

und ***N*** gleich der Vereinigung der genutzten Variablen aller Aktionen in Σ′:

$$N = \bigcup_{\sigma' \in \Sigma'} vars(\sigma') \tag{3.60}$$

Für diese Konstellation betrachten wir nun die verschiedenen Stufen der Interferenz anhand konkreter Beispiele.

Keine Variablen-Interferenz

Bei fehlender Variablen-Interferenz weist keine Aktion aus der Menge Σ′ einer von Aktion ***σ*** veränderten Variablen einen Wert zu. Als Beispiel für fehlende Variablen-Interferenz können wir ***Programm 1*** aus Abbildung 3.4 heranziehen: Hier gilt die geforderte Eigenschaft für ***σ*** = `p1aktion1` und Σ′ = {`p1aktion2`}:

$$cvars(\texttt{p1aktion1}) \cap vars(\texttt{p1aktion2}) = \{a\} \cap \{b\} = \varnothing \tag{3.61}$$

In ***Programm 1*** gibt es also für keine Aktion Variablen-Interferenz mit einer beliebigen Menge anderer Aktionen.

Da in Programm 1 keine Aktion einer Variablen den Initialwert zuweist, entspricht die Variablenmenge jeder Aktion gerade der Menge der veränderten Variablen. Insbesondere gilt hier also auch: ***cvars***(`p6aktion1`) ∩ ***vars***(`p6aktion2`) = ***vars***(`p6aktion1`) ∩ ***vars*** (`p6aktion2`).

Schwache Variablen-Interferenz

Bei der schwachen Variablen-Interferenz operieren verschiedene Aktionen zum Teil auf gleichen Variablen.

Als Beispiel für schwache Variablen-Interferenz betrachten wir ***Programm 6*** aus Abbildung 3.16. Hier agieren die beiden Aktionen `p6aktion1` und `p6aktion2` beide auf der Variablen ***b***. Die Variablen ***a*** und c hingegen werden jeweils exklusiv von `p6aktion1` beziehungsweise `p6aktion2` genutzt.

Es gilt daher für ***σ*** = `p6aktion1` und Σ′ = {`p6aktion2`}:

$$cvars(\texttt{p6aktion1}) \cap vars(\texttt{p6aktion2}) = \{a, b\} \cap \{b, c\} \tag{3.62}$$

$$= \{b\} \tag{3.63}$$

$$\neq \varnothing \tag{3.64}$$

Das bedeutet, dass zwischen den beiden Aktionen `p6aktion1` und `p6aktion2` schwache Variablen-Interferenz herrscht.

Starke Variablen-Interferenz

Im Fall der starken Variablen-Interferenz werden alle veränderten Variablen, mit denen Aktion ***σ*** arbeitet, auch durch andere Aktionen genutzt. Betrachten wir zur

Illustration einen einfachen Fall mit ***M*** = ***cvars***(***σ***) und $N = \bigcup_{\sigma' \in \Sigma'} vars(\sigma')$ für $\Sigma' = \{\sigma', \sigma''\}$. Abbildung 3.22 zeigt schematisch die vollständige Überlappung der Menge der veränderten Variablen der Aktion ***σ*** durch die Vereinigung der beiden Variablenmengen der Aktionen ***σ′*** und ***σ′′***.

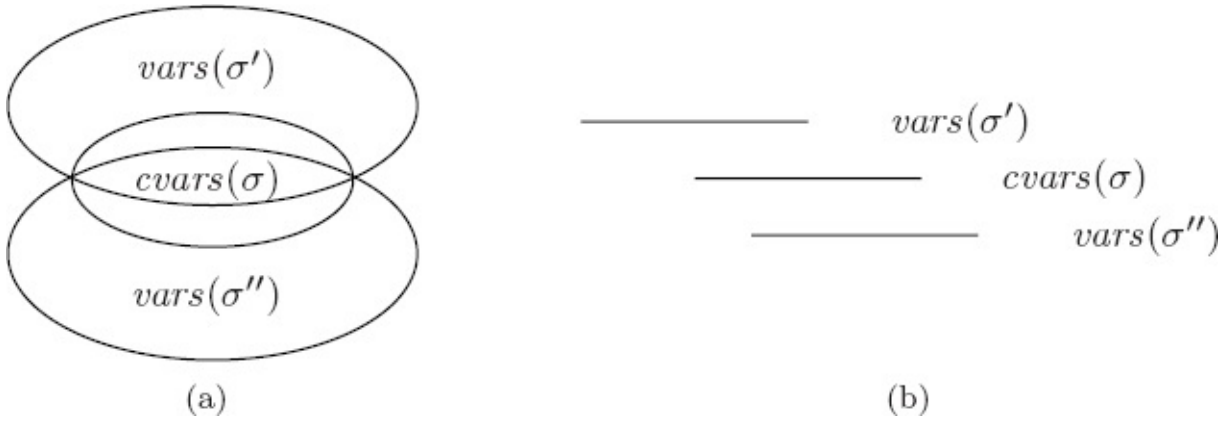


Abbildung 3.22: Schematische Darstellung der Variablenmengen bei ***starker Variablen-Interferenz***.

Einen Spezialfall der starken Variablen-Interferenz stellt der Fall dar, in dem die überlagernde Teilmenge Σ′ ⊂ Σ nur eine Aktion ***σ′*** enthält, also die Mächtigkeit Eins hat: |Σ′| = 1. In diesem Fall gibt es eine einzelne Aktion, die alle von Aktion ***σ*** veränderten Variablen ebenfalls nutzt. In diesem Fall gelten ***M*** = ***cvars***(***σ***) und ***N*** = ***vars***(***σ′***), und die Überschneidung lässt sich wie im Falle der allgemeinen Interferenz bereits in Abbildung 3.20 gezeigt darstellen.

Mit ***Programm 3*** ist ein Beispiel für starke Variablen-Interferenz gegeben, da Aktion `p3aktion1` lediglich der Variablen ***a*** den Wert 1 zuweist und Aktion `p3aktion2` sowohl der Variablen ***a*** als auch ***b*** einen Wert zuweist.

Absolute Variablen-Interferenz

Im Falle absoluter Variablen-Interferenz sind die Menge der veränderten Variablen von Aktion ***σ*** und der Vereinigung der Variablenmengen aller Aktionen in Σ′ gleich.

Zur Illustration wählen wir wieder das Beispiel Σ′ = {***σ′***, ***σ′′***}. Abbildung 3.23 zeigt schematisch die vollständige wechselseitige Überlappung der Menge der veränderten Variablen der Aktion ***σ*** und der Vereinigung der beiden

Variablenmengen der Aktionen ***σ′*** und ***σ′′***. Hier wird auch die Stärke der eindimensionalen Darstellungsweise der Mengen im rechten Teil der Abbildung klar: Während der linke Teil (a) nur erahnen lässt, wie sich die einzelnen Mengen überschneiden, kann dies im Teil (b) leicht nachvollzogen werden. Beide Teile der Abbildung sind daher stets zusammen zu betrachten. Auf die häufig verwendete Kennzeichnung der einzelnen Mengen durch eine unterschiedliche Schraffierung möchten wir zugunsten einer klaren und einheitlichen Darstellung bewusst verzichten.

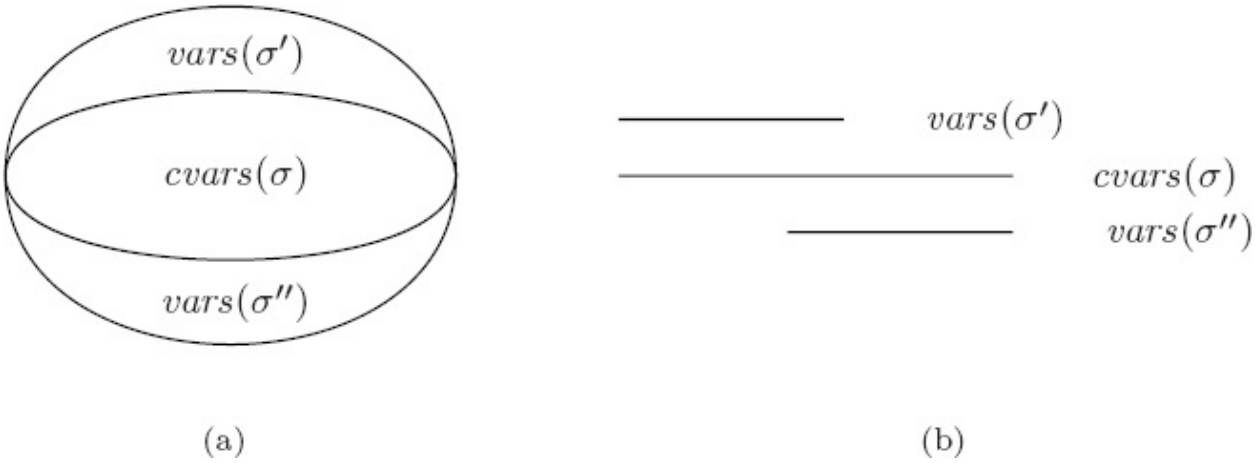


Abbildung 3.23: Schematische Darstellung der Variablenmengen bei ***absoluter Variablen-Interferenz***.

Analog zur starken Interferenz existiert auch für die absolute Variablen-Interferenz wiederum der Spezialfall der Existenz einer einelementigen Menge Σ′, welche die Bedingung für absolute Interferenz erfüllt.

Beispiel 22 (absolute Variableninterferenz) ***Als Beispiel für absolute Variablen-Interferenz sei nun*** Programm 7 ***in Abbildung 3.24 und Abbildung 3.25 gegeben. Hier operieren die beiden Aktionen*** `p7aktion1` ***und*** `p7aktion2` ***ausschließlich auf der Variablen a. Es gilt offensichtlich für σ =*** `p7aktion1` ***und σ′ =*** `p7aktion2` ***absolute Va-riablen-Interferenz:***

$$cvars(\texttt{p7aktion1}) = \{a\} = vars(\texttt{p7aktion2}) \tag{3.65}$$

Variablen: {***a, b***}
Initialzustand: {***a*** = 0 ***b*** = 0}

Aktionen:

`p7aktion1`: $\boldsymbol{a} = 0 \quad \rightarrow \quad \boldsymbol{a} := 1$

`p7aktion2`: $\boldsymbol{b} = 0 \quad \rightarrow \quad \boldsymbol{a} := 1$

Abbildung 3.24: ***Programm 7***.

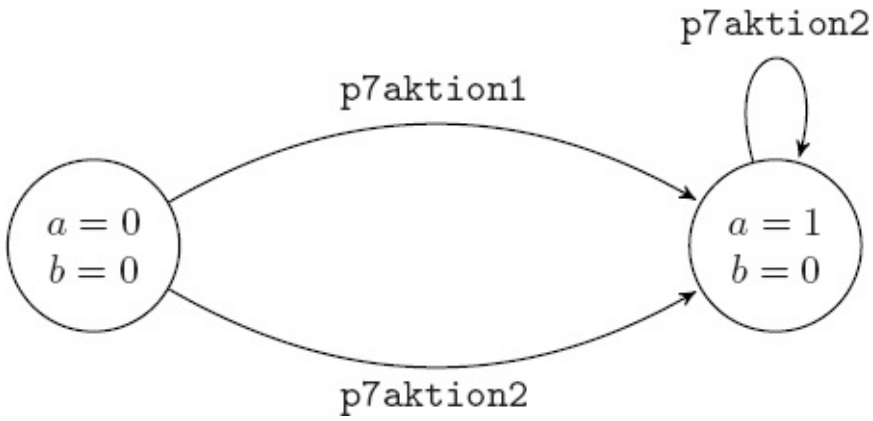


Abbildung 3.25: Zustandsübergangsdiagramm von ***Programm 7***.

Wir beobachten, dass wir im Zustand $\{\boldsymbol{a} = 0, \boldsymbol{b} = 0\}$ ***aufgrund der gemeinsamen charakteristischen Kontraspur*** $\{\{\boldsymbol{a} = 0\}\}$ ***die Aussage treffen können, dass keine der beiden Aktionen stattgefunden hat, im Zustand*** $\{\boldsymbol{a} = 1, \boldsymbol{b} = 0\}$ ***jedoch nicht rekonstruieren können, welche Aktion ausgeführt wurde. Es können sogar beide Aktionen ausgeführt worden sein, falls zuerst Aktion*** `p7aktion1` ***ausgeführt wurde und anschließend Aktion*** `p7aktion2`.

3.6.3 Spuren-Interferenz

Nun verfeinern wir die Variablen-Interferenz, in dem wir nicht die Interferenz hinsichtlich der von einer Aktion benutzten Variablen betrachten, sondern auch die jeweils durch eine Aktion zugewiesenen Werte. Wir betrachten also Interferenz von Mengen von Spuren:

Sei Σ die Menge aller Aktionen eines Systems $\boldsymbol{S}$ wie zuvor und seien $\boldsymbol{\sigma} \in \Sigma$ eine Aktion dieses Systems und $\Sigma' \subset \Sigma$ eine Menge anderer Aktionen ($\boldsymbol{\sigma} \notin \boldsymbol{\Sigma'}$). Dann setzen wir $\boldsymbol{M}$ gleich der Spurenmenge der Aktion $\boldsymbol{\sigma}$:

$$M = E(\sigma) \tag{3.66}$$

und $\boldsymbol{N}$ gleich der Vereinigung der Spurenmengen aller Aktionen in Σ':

$$N = \bigcup_{\sigma' \in \Sigma'} E(\sigma') \tag{3.67}$$

Für diese Konstellation betrachten wir nun wiederum die verschiedenen Stufen der Interferenz anhand konkreter Beispiele.

Keine Spuren-Interferenz

Bei fehlender Spuren-Interferenz weist keine Aktion aus der Menge Σ′ einer Variablen dieselben Werte zu wie Aktion ***σ***. Daher gibt es keine Überschneidung der Spurenmengen beider Aktionen (abgesehen von der leeren Menge).

Als Beispiel für die Nicht-Interferenz können wir wiederum ***Programm 1*** aus Abbildung 3.4 heranziehen: Hier gilt die geforderte Eigenschaft wechselseitig für beide Aktionen:

$$E(\texttt{p1aktion1}) \cap E(\texttt{p1aktion2}) = \{\{a = 1\}, \varnothing\} \cap \{\{b = 1\}, \varnothing\} = \{\varnothing\} \tag{3.68}$$

In ***Programm 1*** gibt es also keine Spuren-Interferenz.

Schwache Spuren-Interferenz

Das Vorhandensein von Interferenz bedeutet, dass es im Gegensatz zur Nicht-Interferenz Aktionen gibt, deren Spuren sich zumindest teilweise überlagern.

Als Beispiel für schwache Interferenz betrachten wir ***Programm 6***. Hier agieren die beiden Aktionen `p6aktion1` und `p6aktion2` beide auf der Variablen ***b*** und weisen ihr den Wert 1 zu. Die Variablen ***a*** und ***c*** hingegen werden jeweils exklusiv von `p6aktion1` beziehungsweise `p6aktion2` genutzt. Es gilt für ***σ*** =`p6aktion1` und Σ′ = `{p6aktion2}`:

$$\begin{aligned} E(\texttt{p6aktion1}) \cap E(\texttt{p6aktion2}) = {} & \{\{a = 1\}, \{b = 1\}, \{a = 1, b = 1\}, \varnothing\} \cap \\ & \{\{b = 1\}, \{c = 1\}, \{b = 1, c = 1\}, \varnothing\} \end{aligned} \tag{3.69}$$

$$= \{\{b = 1\}, \varnothing\} \tag{3.70}$$

$$\neq \{\varnothing\} \tag{3.71}$$

Das bedeutet, dass zwischen den beiden Aktionen `p6aktion1` und `p6aktion2` schwache Spuren-Interferenz herrscht.

Starke Spuren-Interferenz

Im Fall der starken Spuren-Interferenz werden die Spuren einer Aktion $\boldsymbol{\sigma}$ sogar vollständig durch andere Aktionen überlagert.

Einen Spezialfall der starken Interferenz stellt wiederum der Fall dar, in dem die überlagernde Teilmenge $\Sigma' \subset \Sigma$ nur eine Aktion $\boldsymbol{\sigma'}$ enthält, also die Mächtigkeit 1 hat: $|\Sigma'| = 1$. In diesem Fall gibt es eine einzelne Aktion, die alle von Aktion $\boldsymbol{\sigma}$ erzeugten Spuren ebenfalls erzeugt ($\boldsymbol{M} = \boldsymbol{E}(\boldsymbol{\sigma})$ und $\boldsymbol{N} = \boldsymbol{E}(\boldsymbol{\sigma'})$).

Programm 3 ist ein Beispiel für starke Spuren-Interferenz, da Aktion `p3aktion1` der Variablen ***a*** den Wert 1 zuweist und Aktion `p3aktion2` den Wert 1 sowohl der Variablen ***b***, als auch ***a*** zuweist.

Absolute Spuren-Interferenz

Im Falle der absoluten Spuren-Interferenz werden nicht nur die Spuren einer Aktion überlagert, sondern die überlagerte Aktion überdeckt gleichzeitig auch die Spuren der überlagernden Aktionen. Das bedeutet, dass gleichermaßen die Spuren jeder einzelnen Aktion der Menge Σ' vollständig von denen der Aktion $\boldsymbol{\sigma}$ überlagert werden.

Analog zur starken Interferenz existiert wiederum der Spezialfall der Existenz einer einelementigen Menge Σ', welche die Bedingung für absolute Interferenz erfüllt.

Als Beispiel für die absolute Spuren-Interferenz betrachten wir wiederum das ***Programm 7*** aus Abbildung 3.24 und dem Zustandsübergangsdiagramm in Abbildung 3.25. Hier weisen beide Aktionen `p7aktion1` und `p7aktion2` der Variablen ***a*** den Wert 1 zu. Es gilt daher für $\boldsymbol{\sigma}$ =`p7aktion1` und $\boldsymbol{\sigma'}$ =`p7aktion2` absolute Interferenz:

$$E(\texttt{p7aktion1}) = \{\{a = 1\}, \varnothing\} = E(\texttt{p7aktion2}) \tag{3.72}$$

3.6.4 Kombinationen von Interferenz

In den obigen Beispielen zur Interferenz wurde mit Bedacht für eine bestimmte Stufe der Variablen- und Spuren-Interferenz jeweils das gleiche Beispielprogramm herangezogen. Es lag also immer strikt die gleiche Stärke der Variablen- und der Spuren-Interferenz vor. Wir bezeichnen dies als ***symmetrische Kombinationen*** der Interferenz. Wir bezeichnen diese im Folgenden mit ***symmetrisch fehlender Interferenz, symmetrisch schwacher Interferenz, symmetrisch starker Interferenz*** und ***symmetrisch absoluter Interferenz. Symmetrisch schwache Interferenz*** gibt also beispielsweise den Fall ***strikt schwacher Variablen-Interferenz*** bei ***strikt schwacher Spuren-Interferenz*** an. Alle anderen Kombinationen von Interferenz, wie beispielsweise schwache Spuren-Interferenz bei starker Variablen-Interferenz bezeichnen wir im Folgenden als ***asymmetrischen Kombinationen*** der Interferenz. Diese spielen aber bei den weiteren Betrachtungen eine untergeordnete Rolle und werden daher an dieser Stelle nicht weiter ausgeführt.

3.6.5 Interferierende Aktionen, Überdeckung und Äquivalenz

Für die Untersuchung von Kriterien für die Rekonstruierbarkeit ausgeführter Aktionen spielen die zuvor definierten Stufen der Interferenz eine zentrale Rolle. Neben der Stärke der Interferenz ist weiterhin die Menge der für die Interferenz verantwortlichen Aktionen relevant. Die Menge der interferierenden Aktionen ließe sich nun gleichermaßen bezüglich der Variablen-Interferenz und bezüglich der Spuren-Interferenz definieren. Da wir diese Menge im Folgenden jedoch ausschließlich für die Variablen-Interferenz als stärkere Form der Interferenz (und nicht auf der Verfeinerung durch Spuren) benötigen, werden wir die Menge interferierender Aktionen lediglich bezüglich der Variablen-Interferenz einführen.[3]

Definition 19 (Interferierende Aktionen) ***Wir definieren die*** Menge der interferierenden Aktionen ***(Interfering Actions***, I***) einer Aktion σ bezüglich einer Menge anderer Aktionen*** Σ′ ***als I***(***σ***, Σ′) ***wie folgt:***

$$I(\sigma, \Sigma') = \{\sigma' \in \Sigma' | cvars(\sigma) \cap vars(\sigma') \neq \varnothing\} \tag{3.73}$$

Die so definierte Menge der interferierenden Aktionen einer Aktion bezüglich einer Menge anderer Aktionen ist genau dann nicht leer, wenn (zumindest)

schwache Variablen-Interferenz zwischen dieser Aktion und der Menge der anderen Aktionen herrscht.

Neben der Menge interferierender Aktionen ist es wichtig, diejenigen Aktionen zu identifizieren, ohne welche lediglich eine schwächere Stufe der Interferenz erreicht wird. Hierzu betrachten wir zunächst eine Menge von interferierenden Aktionen $\tilde{\Sigma}$, welche zusammen in allen von σ veränderten Variablen interferieren und die Menge der veränderten Variablen von σ damit überdecken. Allerdings kann es mehrere solcher Mengen geben, so dass sich eine solche nicht eindeutig bestimmen lässt. Weiterhin werden wir im Folgenden ***alle*** möglichen derartigen Überdeckungen berücksichtigen. Aus diesem Grund definieren wir hier die Menge aller solcher Überdeckungen als ***Uberdeckungsmenge***[4]. Wir konstruieren diese Menge wie folgt: Wir betrachten alle möglichen Teilmengen der interferierenden Aktionen, also die Potenzmenge $\boldsymbol{P}$ von $\boldsymbol{I}(\sigma, \Sigma')$. Aus dieser Menge wählen wir diejenigen Kombinationen von Aktionen aus, bei denen die Vereinigung der Variablenmengen die Menge der veränderten Variablen von σ vollständig enthält.

Definition 20 (Überdeckungsmenge) ***Wir definieren die*** Überdeckungsmenge (***Covering Set***, CS) ***einer Aktion σ bezüglich einer Menge anderer Aktionen*** Σ' $\boldsymbol{CS}(\sigma, \Sigma')$ ***als die Menge aller Mengen von Aktionen, deren Variablenmengen zusammen alle veränderten Variablen von σ enthalten. Formal gilt:***

$$CS(\sigma, \Sigma') = \{\tilde{\Sigma} \in \mathcal{P}(I(\sigma, \Sigma')) : cvars(\sigma) \subseteq \bigcup_{\tilde{\sigma} \in \tilde{\Sigma}} vars(\tilde{\sigma})\} \tag{3.74}$$

Die so definierte Überdeckungsmenge einer Aktion bezüglich einer Menge anderer Aktionen ist wiederum genau dann nicht leer, wenn (zumindest) ***starke Variablen-Interferenz*** zwischen dieser Aktion und der Menge der anderen Aktionen existiert.

Hiermit sind also solche Kombinationen interferierender Aktionen identifiziert, welche in Kombination miteinander dazu geeignet sind, alle von σ erzeugten Spuren auszulöschen. Falls eine solche Kombination existiert, so erfüllt jedoch beispielsweise auch die Menge aller Aktionen diese Eigenschaft. Das bedeutet insbesondere, dass die Überdeckungsmenge nicht notwendigerweise minimal ist. Als Beispiel betrachten wir Abbildung 3.26 auf der nächsten Seite. Die Abbildung zeigt schematisch die Menge der veränderten Variablen einer Aktion σ, welche durch die Vereinigung der Variablenmengen der Aktionen σ', σ'' und σ''' überlagert wird. Offensichtlich ist diese

Überlagerung jedoch nicht minimal, da bereits die Vereinigung der Variablenmengen der Aktionen σ'' und σ''' eine Überlagerung darstellen.

Aus diesem Grund schließen wir weiterhin solche Mengen von Aktionen aus, für die eine echte Teilmenge existiert, welche ebenfalls die gesamte veränderte Variablenmenge der Aktion σ erzeugt. Die so resultierende Menge von minimalen Mengen überdeckender Aktionen bezeichnen wir als ***minimale Überdeckungsmenge***.

Definition 21 (Minimale Überdeckungsmenge) ***Wir definieren die*** Minimale Überdeckungsmenge (***Minimal Covering Set***, MCS) ***einer Aktion σ bezüglich einer Menge anderer Aktionen*** Σ' ***als MCS***(σ, Σ') ***wie folgt:***

$$MCS(\sigma, \Sigma') = \{\tilde{\Sigma} \in CS(\sigma, \Sigma') | \forall \sigma' \in \tilde{\Sigma} : cvars(\sigma) \nsubseteq \bigcup_{\tilde{\sigma} \in \tilde{\Sigma} \setminus \{\sigma'\}} vars(\tilde{\sigma})\} \tag{3.75}$$

In einem letzten Schritt lässt sich die Definition der Überdeckungsmenge auf den Fall der ***absoluten Variablen-Interferenz*** übertragen, indem statt der Eigenschaft, dass die Menge der veränderten Variablen der betrachteten Aktion σ Teilmenge einer Vereinigung von Variablenmengen ist, nun Gleichheit gefordert wird. Das bedeutet, dass die Vereinigung der Variablenmengen der interferierenden Aktionen genau die Variablenmenge von σ erzeugt. Die Menge solcher Mengen von Aktionen bezeichnen wir als ***Äquivalenzmenge***.

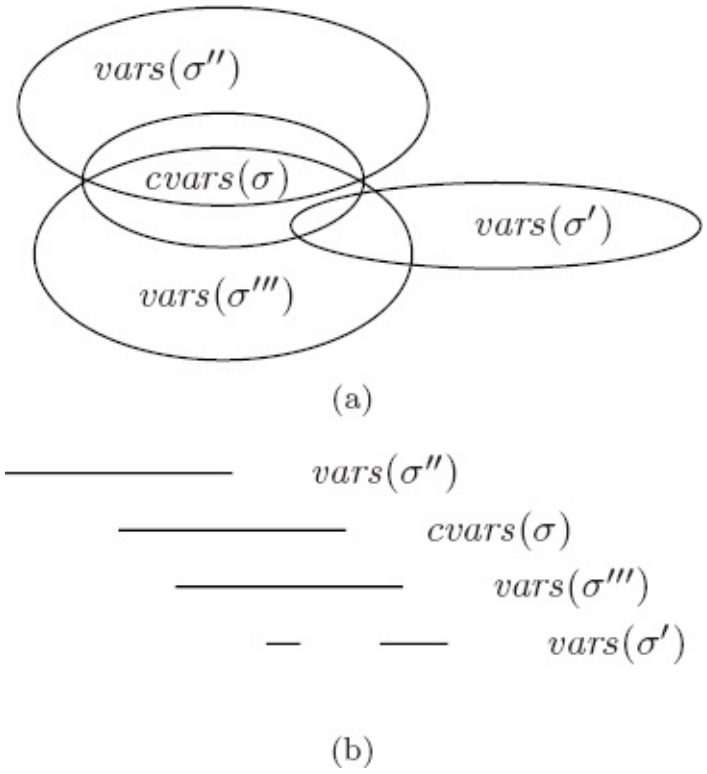


Abbildung 3.26: Schematische Darstellung einer nicht minimalen Überdeckung der Variablenmenge von Aktion ***σ***.

Definition 22 (Äquivalenzmenge) ***Wir definieren die*** Äquivalenzmenge (***Equivalent Set***, ES) ***einer Aktion σ bezüglich einer Menge anderer Aktionen*** Σ' ***als ES***(σ, Σ') ***wie folgt:***

$$ES(\sigma, \Sigma') = \{\tilde{\Sigma} \in \mathcal{P}(I(\sigma, \Sigma')) | cvars(\sigma) = \bigcup_{\tilde{\sigma} \in \tilde{\Sigma}} vars(\tilde{\sigma})\} \tag{3.76}$$

Die so definierte Äquivalenzmenge einer Aktion bezüglich einer Menge anderer Aktionen ist genau dann nicht leer, wenn ***absolute Variablen-Interferenz*** zwischen dieser Aktion und der Menge der anderen Aktionen existiert.

Hiermit sind also solche Kombinationen interferierender Aktionen identifiziert, welche in Kombination miteinander genau die Menge der von ***σ*** veränderten Variablen erzeugen. Analog zur Definition der ***Minimalen Überdeckungsmenge*** muss ein solches Element der Äquivalenzmenge jedoch nicht notwendigerweise minimal sein.

Aus diesem Grund schließen wir wiederum solche Mengen von Aktionen aus, für die eine echte Teilmenge existiert, welche ebenfalls genau die Menge der veränderten Variablen der Aktion ***σ*** erzeugt. Die so resultierende Menge bezeichnen wir als ***minimale Äquivalenzmenge***.

Definition 23 (Minimale Äquivalenzmenge) ***Wir definieren die*** minimale Äquivalenzmenge (***Minimal Equivalent Set***, MES***) einer Aktion σ bezüglich einer Menge anderer Aktionen*** Σ′ ***als MES***(***σ***, Σ′) ***wie folgt:***

$$MES(\sigma, \Sigma') = \{\tilde{\Sigma} \in ES(\sigma, \Sigma') | \forall \sigma' \in \tilde{\Sigma} : cvars(\sigma) \neq \bigcup_{\tilde{\sigma} \in \tilde{\Sigma} \setminus \{\sigma'\}} vars(\tilde{\sigma})\} \tag{3.77}$$

Die minimale Äquivalenzmenge enthält also ausschließlich solche Überdeckungen, die minimal sind. Die Minimalität wird dadurch sichergestellt, dass nur solche Überdeckungen in der minimalen Äquivalenzmenge behalten werden, die keine vollständige Überdeckung mehr erzeugen, wenn auch nur eine Aktion entfernt wird.

3.7 Rekonstruktion

In diesem Abschnitt widmen wir uns nun der Frage nach den Zusammenhängen der im vorherigen Abschnitt erörterten Interferenz und der Lösbarkeit unserer in Abschnitt 3.3 formulierten Rekonstruktionsprobleme. Wir betrachten hierzu zunächst allgemeine Zusammenhänge zwischen der Rekonstruierbarkeit von Ereignissen und der Interferenz. Danach untersuchen wir hinreichende Bedingungen für die Lösbarkeit des spezifischen Rekonstruktionsproblems (SRP) und des spezifischen Gruppen-Rekonstruktionsproblems (SGRP). Schließlich fassen wir die Ergebnisse der einzelnen Lemmata in fünf Theoremen zusammen. Die Beweise haben wir in diesem Werk aus Platzgründen weggelassen, sie können aber bei Interesse andernorts (Dewald, 2012) nachgelesen werden.

3.7.1 Das zugrundeliegende System

In den folgenden Lemmata und Theoremen beziehen wir uns immer auf ein System $\boldsymbol{S} = (\boldsymbol{V}, \Sigma, \boldsymbol{q}_0)$ wie zuvor definiert.

Weiterhin werden wir häufig den Begriff ***aller anderen Aktionen*** in Bezug auf eine konkrete Aktion σ benötigen, weshalb wir diesen der Einfachheit halber an dieser Stelle definieren:

Definition 24 (Alle anderen Aktionen) ***Seien*** $S = (V, \Sigma, q_0)$ ***eine System und*** $\sigma \in \Sigma$ ***eine Aktion dieses Systems. Dann definieren wir bezüglich*** σ ***die*** Menge aller anderen Aktionen Σ' ***als die Menge aller Aktionen*** Σ ***ohne die Aktion*** σ ***selbst. Wenn im Folgenden*** Σ' ***nicht explizit definiert ist, gelte also immer*** $\Sigma' = \Sigma \setminus \{\sigma\}$.

Da auch der Begriff der von einer Aktion ***exklusiv genutzten Variablen*** an verschiedenen Stellen zur Beweisführung benötigt wird, definieren wir diesen hier ebenfalls:

Definition 25 (Exklusiv genutzte Variablen) ***Seien*** $S = (V, \Sigma, q_0)$ ***ein System,*** $\sigma \in \Sigma$ ***eine Aktion dieses Systems und*** Σ' ***eine Menge anderer Aktionen. Dann definieren wir die Menge der von*** σ ***in Bezug auf*** Σ' exklusiv genutzten Variablen (***exclusively used variables) exvars***(σ, Σ') ***als die Menge aller von*** σ ***veränderten Variablen, in denen keine Variablen-Interferenz mit anderen Aktionen aus*** Σ' ***besteht:***

$$exvars(\sigma, \Sigma') = \{v \in cvars(\sigma) | \forall \sigma' \in' : v \notin vars(\sigma')\}$$

Unter Verwendung dieser Definitionen formulieren wir nun unsere ersten Beobachtungen hinsichtlich der Lösbarkeit des spezifischen Rekonstruktionsproblems.

3.7.2 Lösbarkeit des SRP

In diesem Abschnitt befassen wir uns mit der Lösbarkeit des spezifischen Rekonstruktionsproblems (SRP) für das System S. Wir betrachten zunächst die Lösbarkeit dieses Problems für einen gegebenen Zustand, und anschließend untersuchen wir, welche Aussagen zur Lösbarkeit über das gesamte System möglich sind.

Lösbarkeit des SRP und charakteristische (Kontra-)Spuren

Zunächst betrachten wir Voraussetzungen für die Rekonstruierbarkeit von Ereignissen in einem konkreten Zustand anhand charakteristischer Spuren und Kontraspuren. Später untersuchen wir dann für das gesamte System Kriterien für das Vorhandensein solcher Spuren. Hierzu stellen wir nun einige grundlegende Lemmata auf, welche die wichtigsten Zusammenhänge fixieren. Die weiteren Einsichten folgen dann relativ einfach unter Anwendung dieser Lemmata. Wir beginnen mit der Bedeutung charakteristischer Spuren für die Rekonstruktion:

Lemma 1 ***Wenn in einem Zustand q*** charakteristische Spuren ***einer Aktion σ*** $\in \Sigma$ ***bezüglich einer Menge von Aktionen*** $\tilde{\sum} \subseteq \Sigma$ ***beobachtet werden können und kein Pfad α*** $\in A(q)$ ***zu Zustand q existiert, auf dem eine andere Aktion als die Aktion σ oder beliebige Aktionen aus*** $\tilde{\sum}$ ***(also eine Aktion σ′*** $\in \Sigma''$ ***für*** $\Sigma'' = \Sigma \setminus (\{\sigma\} \cup \tilde{\sum})$ ***ausgeführt wurden, dann existiert kein Pfad zu q, auf dem die Aktion σ nicht ausgeführt wurde***.

Setzen wir nun die Beobachtung charakteristischer Spuren bezüglich der Menge ***aller anderen Aktionen*** anstatt einer beliebigen Menge von Aktionen voraus, so gilt die Behauptung sogar ohne zusätzliche Bedingungen an den betrachteten Zustand:

Lemma 2 ***Wenn in einem Zustand q*** charakteristische Spuren ***einer Aktion σ*** $\in \Sigma$ ***bezüglich der Menge*** aller anderen Aktionen Σ' ***beobachtet werden können, so existiert in diesem System kein Pfad α*** $\in A(q)$ ***zu diesem Zustand, auf dem Aktion σ nicht ausgeführt wurde***.

Als Nächstes führen wir die analoge Betrachtung in Bezug auf charakteristische ***Kontraspuren*** durch, um zu zeigen, wann die Beobachtung charakteristischer Kontraspuren es erlaubt, die Ausführung einer Aktion definitiv auszuschließen:

Lemma 3 ***Wenn in einem Zustand q*** charakteristische Kontraspuren ***einer Aktion σ*** $\in \Sigma$ ***bezüglich einer Menge von Aktionen*** $\tilde{\sum} \subseteq \Sigma$ ***beobachtet werden können und kein Pfad α*** $\in A(q)$ ***zu Zustand q existiert, auf dem eine andere Aktion als die Aktion σ oder beliebige Aktionen aus*** $\tilde{\sum}$ ***(also eine Aktion σ′*** $\in \Sigma'$ ***′ für*** $\Sigma'' = \Sigma \setminus (\{\sigma\} \cup \tilde{\sum})$ ***ausgeführt wurden, dann existiert kein Pfad zu q, auf dem die Aktion ausgeführt wurde***.

Erweitern wir die Voraussetzung auf die Beobachtung charakteristischer Spuren bezüglich der Menge ***aller anderen Aktionen***, so folgt wiederum die Behauptung ohne Einschränkungen des betrachteten Zustandes:

Lemma 4 ***Wenn in einem Zustand q*** charakteristische Kontraspuren ***einer Aktion σ*** $\in \Sigma$ ***bezüglich der Menge*** aller anderen Aktionen Σ' ***beobachtet werden können, so existiert in diesem System*** kein ***Pfad α*** $\in A(q)$ ***zu diesem Zustand, auf dem Aktion σ ausgeführt wurde***.

Wir konnten also bisher feststellen, dass sowohl die Beobachtung charakteristischer Spuren als auch charakteristischer Kontraspuren einen entscheidenden Einfluss auf die Lösbarkeit des spezifischen Rekonstruktionsproblems hat. Nun stellt sich die Frage, wie es sich im Falle gleichzeitiger Beobachtung beider Arten von Spuren in ein und demselben Zustand mit der Lösbarkeit dieses Problems verhält. Allerdings drängt sich beinahe intuitiv anhand der Definition dieser konträren Spurenmengen die Vermutung auf, dass ein solcher Zustand nicht eintreten kann. Wir formalisieren diese Vermutung im folgenden Lemma:

Lemma 5 ***Für eine beliebige Aktion σ*** $\in \Sigma$, ***eine beliebige Menge von Aktionen*** $\tilde{\Sigma} \subseteq \Sigma$ ***und einen beliebigen Zustand q gilt: Wenn kein Pfad α*** $\in A(q)$ ***zu q existiert, auf dem eine andere Aktion als die Aktion σ oder beliebige Aktionen aus*** $\tilde{\Sigma}$ ***(also eine Aktion σ'*** $\in \Sigma''$ ***für*** $\Sigma'' = \Sigma \setminus (\{\sigma\} \cup \tilde{\Sigma})$***) ausgeführt wurde, dann können in q nicht sowohl charakteristische Spuren als auch charakteristische Kontraspuren von σ bezüglich*** $\tilde{\Sigma}$ ***beobachtet werden. Formal gilt dann also:***

$$\exists e \in CE(\sigma, \tilde{\Sigma}) : e \subseteq q \Rightarrow \nexists e' \in CXE(\sigma, \tilde{\Sigma}) : e' \subseteq q \tag{3.78}$$

und

$$\exists e \in CXE(\sigma, \tilde{\Sigma}) : e \subseteq q \Rightarrow \nexists e' \in CE(\sigma, \tilde{\Sigma}) : e' \subseteq q \tag{3.79}$$

Die Beobachtungen der vorangegangenen Lemmata fassen wir nun in zwei Theoremen über die Lösbarkeit des SRP zusammen:

Theorem 1 ***Wenn in einem Zustand q charakteristische Spuren oder charakteristische Kontraspuren einer Aktion σ*** $\in \Sigma$ ***bezüglich einer Menge von***

Aktionen $\tilde{\sum} \subseteq \Sigma$ ***beobachtet werden können und kein Pfad*** $\alpha \in A(q)$ ***zu Zustand q existiert, auf dem eine andere Aktion als die Aktion σ oder beliebige Aktionen aus*** $\tilde{\sum}$ ***ausgeführt wurden, dann kann das SRP für die Aktion σ in Zustand q gelöst werden***.

Analog gilt für die Beobachtung charakteristischer Spuren oder Kontraspuren bezüglich der Menge aller anderen Aktionen das folgende Theorem:

Theorem 2 ***Wenn in einem gegebenen Zustand q charakteristische Spuren oder charakteristische Kontraspuren einer konkreten Aktion σ bezüglich der Menge aller anderen Aktionen*** Σ' ***beobachtet werden können, dann kann das SRP für die Aktion σ in Zustand q gelöst werden***.

Die Lösbarkeit des SRP hängt also von der Möglichkeit der Beobachtung charakteristischer Spuren und charakteristischer Kontraspuren ab. Im Folgenden untersuchen wir nun, welchen Einfluss der Grad der Interferenz auf das Vorhandensein solcher Spuren in einem konkreten Zustand hat. Wir beginnen mit dem im Hinblick auf die Rekonstruktion günstigsten Fall, der Nicht-Interferenz und betrachten dann stetig schlechter werdende Bedingungen für die Rekonstruktion mit steigendem Grad der Interferenz.

Lemma 6 ***Wenn für eine Aktion*** $\sigma \in \Sigma$ ***bezüglich der Menge aller anderen Aktionen*** Σ' keine Variablen-Interferenz ***existiert, kann das*** spezifische Rekonstruktionsproblem ***(SRP) für σ in jedem beliebigen Zustand*** $q \in Q$ ***des Systems gelöst werden***.

Wenn unterschiedliche Aktionen also auf gänzlich disjunkten Mengen von Variablen operieren, kann das SRP in jedem Zustand gelöst werden. Im Fall fehlender Interferenz entspricht dies im Wesentlichen auch der intuitiven Erwartung. Weniger intuitiv ist jedoch der Fall stärkerer Stufen der Interferenz. Daher widmen wir uns nun der schwachen Variablen-Interferenz als nächst stärkere Form der Interferenz und zeigen, dass das SRP auch in diesem Fall gelöst werden kann.

Lemma 7 ***Wenn es*** strikt schwache Variablen-Interferenz ***einer Aktion*** $\sigma \in \Sigma$ ***bezüglich der Menge aller anderen Aktionen*** Σ' ***gibt, kann das*** spezifische Rekonstruktionsproblem ***(SRP) für σ in jedem beliebigen Zustand*** $q \in Q$ ***des Systems gelöst werden***.

Für die starke Variablen-Interferenz können wir nicht zeigen, dass das spezifische Rekonstruktionsproblem in jedem Zustand gelöst werden kann. Es ist sogar naheliegend, dass das SRP bei starker Interferenz im Allgemeinen ***nicht*** in jedem Zustand gelöst werden kann, da in diesem Fall Zustände existieren können, in denen weder charakteristische Spuren, noch charakteristische Kontraspuren einer Aktion ***σ*** beobachtet werden können. Solche Zustände sind dann erreicht, wenn für mindestens eine minimale Überdeckung $\tilde{\sum} \in$ ***MCS*(*σ,* Σ′)** alle Aktionen $\boldsymbol{\sigma} \in \tilde{\sum}$ ausgeführt wurden. Jedoch muss die Lösbarkeit des SRP auch unter diesen Voraussetzungen nicht grundsätzlich ausgeschlossen werden.

Auch wenn bei starker oder gar absoluter Variablen-Interferenz einer Aktion bezüglich der Menge aller anderen Aktionen das SRP nicht in ***jedem Zustand*** gelöst werden kann, so kann auch in einem solchen System das SRP unter bestimmten Bedingungen dennoch ***für einzelne Zustände*** gelöst werden, wie wir im Folgenden zeigen.

Lemma 8 ***Wenn für eine Aktion σ ∈ Σ auf keinem zu einem gegebenen Zustand q führenden Pfad α ∈ A(q) alle Aktionen aus einer minimalen Überdeckung*** $\tilde{\sum} \in$ ***MCS*(*σ,* Σ′) *ausgeführt werden (formal:*** $\nexists \alpha \in \mathcal{A}(q) : \exists \tilde{\Sigma} \in MCS(\sigma) : \tilde{\sigma} \in \alpha \; \forall \tilde{\sigma} \in \tilde{\Sigma}$***), kann das*** spezifische Rekonstruktionsproblem ***(SRP) für σ in Zustand q gelöst werden***.

Intuitiv besagt Lemma 8, dass das SRP für eine Aktion (auch bei starker Variablen-Interferenz) gelöst werden kann, wenn alle Kombinationen von Aktionen, welche die gleichen Spuren wie ***σ*** erzeugt haben könnten, ausgeschlossen werden können.

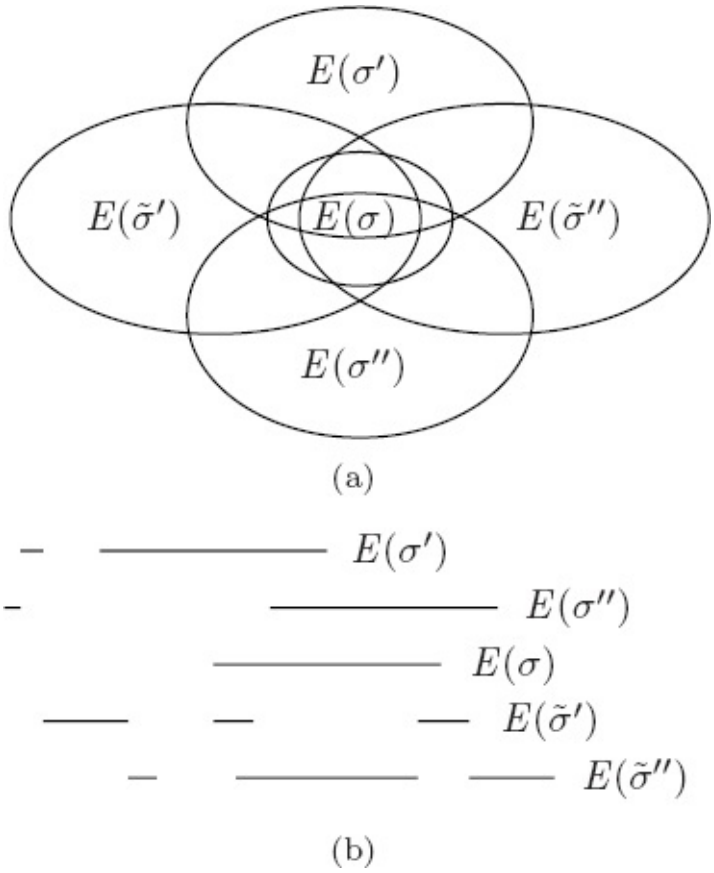


Abbildung 3.27: Schematische Darstellung einer nicht minimalen Überdeckung der Spurenmenge von Aktion ***σ***.

Beispiel 23 (minimale Überdeckungsmenge) ***Wir veranschaufichen dies in Abbildung 3.27. Hier gibt es zwei Mengen von Aktionen, nämlich*** $\{\sigma', \sigma''\}$ ***und*** $\{\tilde{\sigma}', \tilde{\sigma}''\}$, ***deren Vereinigung der Spurenmengen jeweils die Spurenmenge von Aktion σ überlagert. Kann aus jeder dieser beiden Mengen jeweils zumindest eine Aktion ausgeschlossen werden, so erlaubt die Beobachtung von bestimmten Spuren der Aktion σ durchaus die Schlussfolgerung, dass Aktion σ stattgefunden hat***.

Dieses Lemma zeigt, dass selbst wenn für ein System das SRP nicht in jedem Zustand gelöst werden kann, die Möglichkeit besteht, dass in einem konkreten Fall gerade ein solcher Zustand zur Untersuchung vorliegt, in dem das SRP dennoch gelöst werden kann. Die Ergebnisse der vorangegangenen Lemmata zur Lösbarkeit des SRP fassen wir nun in einem Theorem zusammen:

Theorem 3 ***Für eine Aktion σ kann das SRP in einem Zustand q unter den in Tabelle 3.1 genannten Bedingungen gelöst werden. Der angegebene strikte Grad der Variablen-Interferenz bezieht sich hierbei auf die Aktion σ bezüglich der Menge*** Σ'.

Grad Variablen-Interferenz	Bedingung zur Lösbarkeit des SRP
Keine Interferenz	—
Schwache Interferenz	—
Starke Interferenz	$\nexists \alpha \in \mathcal{A}(q) : \exists \tilde{\Sigma} \in MCS(\sigma) : \forall \tilde{\sigma} \in \tilde{\Sigma} : \tilde{\sigma} \in \alpha$
Absolute Interferenz	$\nexists \alpha \in \mathcal{A}(q) : \exists \tilde{\Sigma} \in MCS(\sigma) : \forall \tilde{\sigma} \in \tilde{\Sigma} : \tilde{\sigma} \in \alpha$

Tabelle 3.1: Lösbarkeit des SRP.

Nun betrachten wir den Fall, in dem das SRP nicht gelöst werden kann und stellen fest, dass auch in diesem Fall noch minimale Aussagen über zuvor ausgeführte Aktionen möglich sind, wenn höchstens schwache ***Spuren-Interferenz*** vorliegt.

Lemma 9 ***Wenn für eine Aktion*** $\sigma \in \Sigma$ ***bezüglich der Menge aller anderen Aktionen*** Σ' ***keine*** Spuren-Interferenz ***oder strikt schwache*** Spuren-Interferenz ***existiert, so kann (auch wenn das SRP nicht gelöst werden kann) in jedem Zustand für jede von σ veränderte Variable*** $v \in vars(\sigma)$, ***der keine andere Aktion den gleichen Wert wie die Aktion σ zuweist, festgestellt werden, ob die Aktion σ selbst zuletzt ausgeführt wurde, oder eine der anderen Aktionen***.

Nachdem wir festgestellt haben, dass im Fall starker oder absoluter Variablen-Interferenz das SRP in einem Zustand nur dann gelöst werden kann, wenn für diesen Zustand bestimmte Bedingungen gelten, stellt sich die Frage, welche Aussagen in diesem Fällen hinsichtlich der Rekonstruktion vergangener Ereignisse für ein ***gesamtes*** System getroffen werden können. Wir betrachten hierzu im nächsten Abschnitt hinreichende Bedingungen für die Lösbarkeit des spezifischen Gruppen-Rekonstruktionsproblems.

3.7.3 Lösbarkeit des SGRP

Wir widmen uns nun der Überprüfung hinreichender Bedingungen zur Lösbarkeit des spezifischen Gruppen-Rekonstruktionsproblems (SGRP, siehe Abschnitt 3.3). Wir untersuchen zunächst, welche Bedeutung ***gemeinsame*** charakteristische Spuren einer Menge von Aktionen für die Lösbarkeit des SGRP für diese Menge haben.

Lemma 10 ***Wenn in einem Zustand q*** gemeinsame charakteristische Spuren ***einer Menge von Aktionen*** $\Sigma'' \subseteq \Sigma$ ***bezüglich der Menge*** aller anderen Aktionen $\Sigma' = \Sigma \setminus \Sigma''$ ***beobachtet werden können, so existiert kein Pfad zu q, auf dem nicht zumindest eine Aktion*** $\sigma' \in \Sigma''$ ***ausgeführt wurde***.

Analog betrachten wir nun ebenso die Bedeutung gemeinsamer charakteristischer ***Kontraspuren*** für die Rekonstruktion.

Lemma 11 ***Wenn in einem Zustand q*** gemeinsame charakteristische Kontraspuren einer ***Menge von Aktionen*** $\Sigma'' \subseteq \Sigma$ ***bezüglich der Menge*** aller anderen Aktionen $\Sigma' = \Sigma \setminus \Sigma''$ ***beobachtet werden können, so existiert kein Pfad zu q, auf dem auch nur eine Aktion*** $\sigma' \in \Sigma''$ ***ausgeführt wurde***.

Wie bereits aus dem vorherigen Abschnitt bekannt, können in einem Zustand lediglich ***entweder*** charakteristische Spuren ***oder*** charakteristische Kontraspuren ein und derselben Aktion vorliegen. Die gleiche Eigenschaft gilt auch für ***gemeinsame*** charakteristische Spuren und Kontraspuren, wie wir im folgenden Lemma zeigen:

Lemma 12 ***Für eine beliebige Menge von Aktionen*** $\Sigma'' \subseteq \Sigma$, ***die Menge aller anderen Aktionen*** $\Sigma' = \Sigma \setminus \Sigma''$ ***und einen beliebigen Zustand q gilt: In Zustand q können nicht sowohl gemeinsame charakteristische Spuren als auch gemeinsame charakteristische Kontraspuren von*** Σ'' ***bezüglich*** Σ' ***beobachtet werden. Formal gilt dann also:***

$$\exists e \in CCE(\Sigma'', \Sigma') : e \subseteq q \Rightarrow \nexists e' \in CCXE(\Sigma'', \Sigma') : e' \subseteq q \tag{3.80}$$

und

$$\exists e \in CCXE(\Sigma'', \Sigma') : e \subseteq q \Rightarrow \nexists e' \in CCE(\Sigma'', \Sigma') : e' \subseteq q \tag{3.81}$$

Die Beobachtungen der bisherigen Lemmata dieses Abschnittes fassen wir nun in einem Theorem zusammen:

Theorem 4 ***Wenn in einem Zustand q gemeinsame charakteristische Spuren oder gemeinsame charakteristische Kontraspuren einer Menge von Aktionen*** $\Sigma'' \subseteq \Sigma$ ***bezüglich der Menge*** aller anderen Aktionen $\Sigma' = \Sigma \setminus \Sigma''$ ***beobachtet werden können, so kann das SGRP für die Menge*** Σ'' ***in Zustand q gelöst werden***.

Als Nächstes zeigen wir, dass das SGRP immer lösbar ist, wenn das SRP lösbar ist. Daraufhin zeigen wir jedoch, dass das SGRP weiterhin lösbar sein kann, selbst wenn das SRP nicht gelöst werden kann. Wir geben auch an, für welche Mengen von Aktionen das SGRP dann noch gelöst werden kann.

Lemma 13 ***Wenn das*** spezifische Rekonstruktionsproblem ***(SRP) für eine Aktion*** $\sigma \in \Sigma$ ***in einem Zustand q gelöst werden kann, kann für*** $\Sigma'' = \{\sigma\}$ ***das*** spezifische Gruppen-Rekonstruktionsproblem ***(SGRP) bezüglich der Menge aller anderen Aktionen*** $\Sigma' = \Sigma \setminus \{\sigma\}$ ***gelöst werden***.

Nun widmen wir uns der Situation, in der das SRP nicht gelöst werden kann und zeigen, dass das SGRP trotzdem für eine bestimmte Menge von Aktionen gelöst werden kann.

Lemma 14 ***Wenn in einem Zustand q vom Initialzsutand unterscheidbare Spuren einer Aktion*** $\sigma \in \Sigma$ ***beobachtet werden können (das heißt keine Nullspuren), dann kann, auch wenn das*** spezifische Rekonstruktionsproblem ***(SRP) für σ in q nicht gelöst werden kann, für die Menge***

$$\Sigma'' = \{\sigma\} \cup \boldsymbol{I}(\sigma, \Sigma')$$

das spezifische Gruppen-Rekonstruktionsproblem ***(SGRP) in q gelöst werden***.

Anmerkung: Dies wird offensichtlich um so weniger nützlich, je größer Σ'' wird, insbesondere wenn $\Sigma'' = \Sigma$ gilt (oder annähernd).

Schließlich fassen wir wiederum die Ergebnisse der letzten Lemmata bezüglich der Lösbarkeit des SGRP in einem Theorem zusammen:

Theorem 5 ***Wenn in einem Zustand q vom Initialzsutand unterscheidbare Spuren einer Aktion*** $\sigma \in \Sigma$ ***beobachtet werden können (das heißt keine Nullspuren), kann das SGRP für eine diese Aktion beinhaltende Menge von Aktionen*** $\tilde{\sum}$ ***wie unten in Tabelle 3.2 angegeben in q gelöst werden. Der***

angegebene Grad der Variablen-Interferenz bezieht sich hierbei auf die Aktion σ bezüglich der Menge aller anderen Aktionen Σ'.

Intuitiv bedeutet dieses Theorem: Wenn in einem Zustand vom Initialzustand unterscheidbare Spuren einer Aktion beobachtet werden können, diese jedoch nicht ausreichen, um definitiv auf die Ausführung genau jener Aktion zu schließen, kann immerhin eine Menge von Aktionen gefunden werden, aus der zumindest eine Aktion ausgeführt worden sein muss. Zusammen mit den vorangegangenen Lemmata beschreibt das Theorem weiterhin konstruktiv die Bildung einer für die geltende Stufe der Interferenz minimalen Menge von Aktionen, für welche diese Eigenschaft gilt.

Grad Variablen-Interferenz	Lösbarkeit des SGRP für $\tilde{\Sigma}$
Keine Interferenz	$\tilde{\Sigma} = \{\sigma\}$
Schwache Interferenz	$\tilde{\Sigma} = \{\sigma\}$
Starke Interferenz	$\tilde{\Sigma} = \{\sigma\} \cup I(\sigma, \Sigma')$
Absolute Interferenz	$\tilde{\Sigma} = \{\sigma\} \cup I(\sigma, \Sigma')$

Tabelle 3.2: Lösbarkeit des SGRP.

3.7.4 Lösbarkeit des GRP

Nachdem wir die Lösbarkeit des SRP und des SGRP im Detail untersucht haben, widmen wir uns nun noch einmal unserem zuerst formulierten Problem, dem allgemeinen Rekonstruktionsproblem (GRP), und dessen Lösbarkeit.

Wir beobachten, dass das GRP echt stärker ist als das SRP: Selbst wenn das SRP für alle Aktionen in jedem Zustand gelöst werden kann (was bereits eine sehr mächtige Annahme darstellt), so kann das GRP mit diesem Wissen nicht gelöst werden. Wir zeigen dies anhand eines Beispiels:

Intuitiv könnte man zunächst annehmen, dass aus der Lösbarkeit des SRP für alle Aktionen und alle Zustände eines Systems die Lösbarkeit des GRP für das betreffende System folgt. Man würde in jedem Zustand für jede Aktion das SRP lösen und damit wissen, ob die betreffende Aktion auf allen Pfaden zu diesem Zustand stattgefunden haben muss oder auf keinem und aufgrund dieser Information alle Pfade zu dem gegebenen Zustand beziehungsweise den

ursprünglichen Graphen des Systems zu rekonstruieren suchen. Gelänge dies, so wäre das GRP, welches die Kenntnis aller Pfade zu einem gegebenen Zustand fordert, für das System lösbar. Es lassen sich jedoch Beispiele finden, in denen nicht alle Pfade rekonstruiert werden können, beziehungsweise der auf die beschriebene Art und Weise konstruierte Graph nicht dem ursprünglichen Graphen entspricht. Betrachten wir ***Programm 8*** aus Abbildung 3.28 mit dem Zustandsübergangsdiagramm in Abbildung 3.29.

Beispiel 24 (SRP und GRP) ***In Zustand*** {***a*** = 0, ***b*** = 0} ***wissen wir aufgrund der Lösbarkeit des SRP, dass Aktion*** `p8aktion1` ***definitiv nicht stattgefunden hat. In Zustand*** {***a*** = 1, ***b*** = 0} ***wissen wir entsprechend, dass Aktion*** `p8aktion1` ***definitiv stattgefunden hat. Damit wurden in allen Zuständen alle Aktionen betrachtet und der rekonstruierte Graph würde sich wie in Abbildung 3.30 auf der nächsten Seite gezeigt darstellen. Im Wesentlichen erlaubt die Lösbarkeit des SRP also keine Rückschlüsse auf die Mehrfachausführung von Aktionen, was jedoch eine Voraussetzung für die Lösbarkeit des GRP wäre***.

Variablen: {***a, b***}
Initialzustand: {***a*** = 0, ***b*** = 0}
Aktionen:

`p8aktion1`: ***b*** = 0 → ***a*** := 1

Abbildung 3.28: ***Programm 8***.

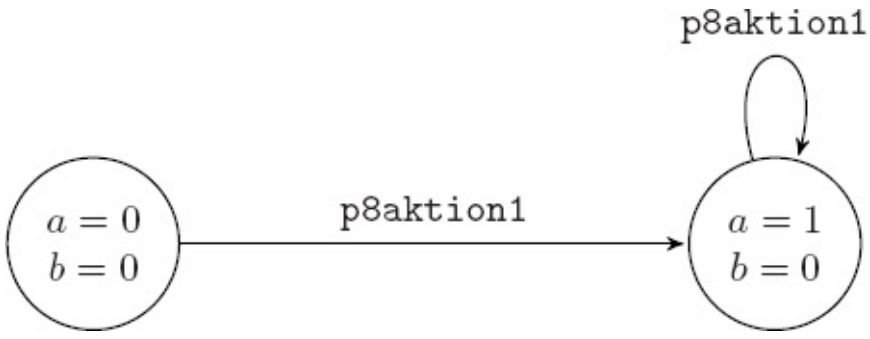


Abbildung 3.29: Zustandsübergangsdiagramm von ***Programm 8***.

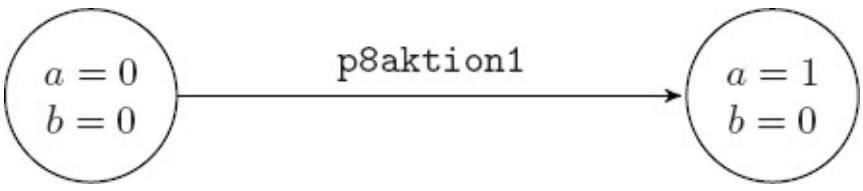


Abbildung 3.30: Rekonstruiertes Zustandsübergangsdiagramm von ***Programm 8***.

Variablen: {***a, b***}
Initialzustand: {***a*** = 0, ***b*** = 0}
Aktionen:
p9aktion1: ***a*** = 0 ∨ ***b*** = 1 → ***a*** := 1
p9aktion2: ***a*** = 1 → ***b*** := 1

Abbildung 3.31: ***Programm 9***.

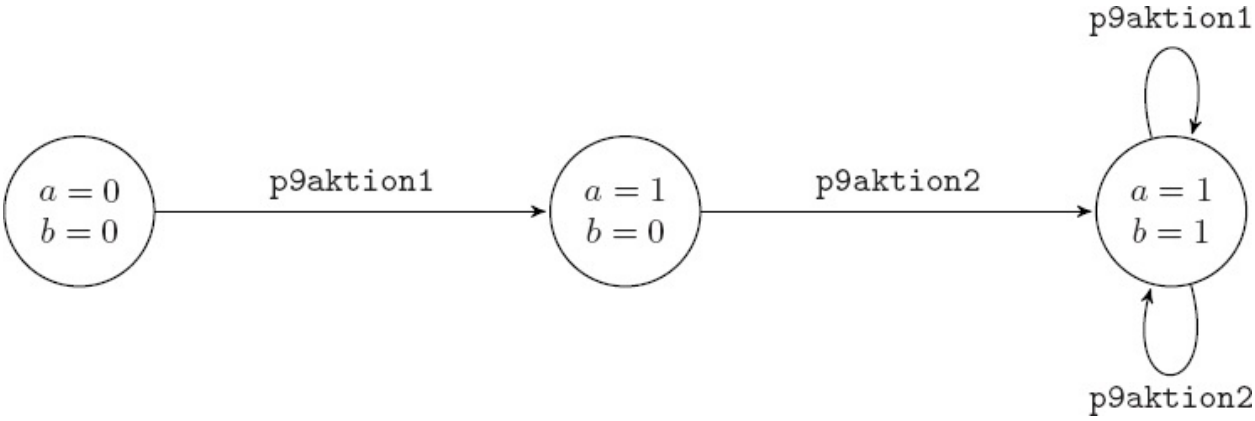


Abbildung 3.32: Zustandsübergangsdiagramm von ***Programm 9***.

Beispiel 25 (Mehrfachausführung und das SRP) ***Das Beispiel von*** Programm 9 ***aus Abbildung 3.31 auf der vorherigen Seite mit dem Zustandsübergangsdiagramm in Abbildung 3.32 illustriert ein entsprechendes Beispiel mit zwei Aktionen. Auch in diesem Beispiel kann die Möglichkeit der Mehrfachausführung der beiden Aktionen `p9aktion1` und `p9aktion2`, welche sich als Schleifen im Zustandsübergangsdiagramm zeigen, durch die Lösbarkeit des SRP nicht detektiert werden***.

Dies zeigt, dass das SRP eine echte Vereinfachung gegenüber dem GRP darstellt. Die Voraussetzung für die Lösbarkeit des GRP erfordert das Wissen über das gesamte untersuchte System. In der Praxis verfügt man jedoch in der Regel nicht über solch umfassendes Wissen. Die Theoreme 3 und 5 zeigen jedoch, dass obgleich das GRP in der Praxis meist nicht gelöst werden kann, aufgrund von Spuren und Kontraspuren doch das SRP und das SGRP, die für eine große Anzahl an Ermittlungsfragen bereits sehr hilfreich sind, gelöst werden können.

3.8 Zusammenfassung

In diesem Kapitel haben wir zunächst, als Verfeinerung der intuitiven Vorstellung eines digitalen Systems als Zustandsautomaten, ein formales Modell eingeführt, welches auf Dijkstras Guarded Commands basiert. Anhand dieses Modells wurde dann die Problemstellung der Assoziation als Kernproblem einer jeden forensischen Untersuchung in drei formale Probleme gefasst. Um Voraussetzungen für die Lösbarkeit dieser Rekonstruktionsprobleme zu untersuchen, haben wir die Begriffe der (charakteristischen und gemeinsamen) Spuren und Kontraspuren sowie die verschiedenen Arten der Interferenz eingeführt. Schließlich konnten wir dann eine Reihe hinreichender Bedingungen für die Lösbarkeit der zuvor definierten Probleme in unterschiedlichen Konstellationen von Interferenz identifizieren und deren Korrektheit beweisen.

Wie eingangs erwähnt, erfolgten alle Betrachtung rein theoretisch (in einem Modell digitaler Systeme). Wenn es dabei heißt, dass eine Aussage „bewiesen“ wurden, heisst das lediglich, dass die Aussage im gegebenen Modell (vermutlich) wahr ist — nicht mehr und nicht weniger. Die aufgestellte Terminologie, also die Begriffe für Spuren, charakteristische Spuren, Kontraspuren und Spureninterferenz sind jedoch auch unter weniger scharfen Randbedingungen sinnvoll und können für die Praxis hilfreich sein.

---

[2]Wir sehen in Abschnitt 3.5.1, dass $\{\boldsymbol{b} = 0\}$ vielmehr Kontraspur von `p1aktion2` ist (alsdazu geeignet ist, die Ausführung der Aktion `p1aktion2` zu widerlegen, anstatt die Ausführung von `p1aktion1` zu belegen).

[3] Zur Definition der analogen Strukturen bezüglich der Spuren-Interferenz muss in den Definitionen diesen Abschnittes lediglich die Funktionen ***vars*** und c***vars*** durch die Funktion ***E*** ersetzt werden, um Spuren anstatt Variablen zugrunde zu legen.

[4] Da es sich bei der Überdeckungsmenge um eine Menge von Mengen handelt, wäre möglicherweise der Begriff der ***Überdeckungsmengenmenge*** angezeigt. Da dieser jedoch umständlich ist und wir im Folgenden immer die Menge ***aller*** Überdeckungen benötigen, verwenden wir hier trotzdem den Begriff der ***Überdeckungsmenge***.

Kapitel 4

Einführung in die Multimediaforensik

Autor: Christian Riess

Dieses Kapitel liefert einen Überblick über den Themenbereich Multimediaforensik. Nach der Bearbeitung dieses Kapitels und der dazugehörigen Aufgaben werden Sie wissen, mit welchen Fragestellungen sich die Multimediaforensik befasst und über welche Methoden sie verfügt. Sie werden in die Lage versetzt, konkrete Problemstellungen aus digitalen Ermittlungen daraufhin zu prüfen, ob Techniken der Multimediaforensik zur Lösung beitragen können. Außerdem werden Sie Erfahrungen mit konkreten Werkzeugen der Multimediaforensik gesammelt haben, die es Ihnen erlauben, die Benutzbarkeit und Anwendbarkeit anderer Werkzeuge kompetent einzuschätzen.

4.1 Themen der Multimediasicherheit

Wie das Gebiet der digitalen Forensik in den Forschungsbereich IT-Sicherheit hineinfällt, so gehört das Gebiet der Multimediaforensik allgemein zum Forschungsbereich der Multimediasicherheit. Der Begriff Multimedia hat seinen Ursprung in der Digitalisierung aller Medien, seien es Bilder, Tondokumente oder Videos. Solche Medien können heutzutage von jedermann erstellt, editiert und ausgetauscht werden. Daraus entsteht ein natürliches Potential für Straftaten im Bereich des Urheberrechts. Insbesondere die redundante Speicherung von Multimediadaten bietet jedoch auch ein reichhaltiges Reservoir für Sicherheitsfragen und -lösungen.

Dieses Kapitel liefert eine Einführung in das Themengebiet der Multimediaforensik und bettet diesen Überblick in den Bereich der Multimediasicherheit ein. Wir werden uns also zunächst beschäftigen mit dem Thema Steganographie, also dem gezielten Verstecken zusätzlicher Information

in Multimediadaten. Anschließend befassen wir uns mit dem Watermarking, also der Markierung des Bildes als Eigentum, und geben abschließend einen Ausblick auf die weiteren Themen dieses Kapitels.

4.1.1 Steganographie: Verdeckte Kommunikation

Kryptographie ist der bekannteste Weg, Daten vor unautorisierter Nutzung zu schützen. Hierbei wird eine Eingabesequenz – zum Beispiel ein Text oder ein Bild – mit Hilfe eines Schlüssels übersetzt in einen Chiffretext. Bei guten Verschlüsselungssystemen lassen die statistischen Eigenschaften des Chiffretextes keinen Rückschluss auf den Inhalt der Eingabe zu. Besonders unauffällig ist diese Art der Kommunikation nicht: Wenn zwei Kommunikationspartner unlesbare Nachrichten austauschen, liegt nahe, dass diese verschlüsselt wurden. In manchen Situationen ist es jedoch wichtig, dass der Kommunikationsvorgang als solcher vor einem Beobachter verborgen wird. Dies kann mit Steganographie erreicht werden. Das Wort stammt ab von den altgriechischen Begriffen „stegos“ und „grafein“ und bedeutet übersetzt in etwa „heimlich schreiben“. Steganographie subsumiert die Methoden, eine private Nachricht unbemerkt über einen öffentlichen (mutmaßlich abgehörten) Kommunikationskanal zu übertragen.

Exkurs 4 (historische Beispiele) ***Ein historisches Beispiel stammt von dem griechischen Tyrannen Histaios, der Herodots Aufzeichnung nach einem Sklaven die Haare abschnitt, und ihm eine geheime Nachricht auf den Kopf tätowierte. Nachdem die Haare nachgewachsen waren, schickte Histaios den Sklaven zu dem Empfänger der Nachricht, der dem Sklaven wiederum den Kopf rasieren ließ (vergleiche Nölle (2009)). Die Geschichte zeigt, dass Steganographie auf sehr unterschiedliche Arten betrieben werden kann***.

Auch in der elektronischen Kommunikation gibt es eine Vielzahl möglicher steganographischer Einbettungen. Beispielsweise arbeitet ein Webserver typischerweise auf zwei Ports, Port 80 für normale Webseiten, und Port 443 für SSL-verschlüsselte Seiten. Dies lässt sich für ein primitives Steganographisches System ausnutzen. Beispielsweise können Serverbetreiber und Webnutzer vereinbaren, dass das Abrufen einer Seite auf Port 80 als „1 “ interpretiert wird, hingegen das Abrufen auf Port 443 als „0“. Somit wird die Übermittlung einer Nachricht an den Serverbetreiber als normales Surfen getarnt.

Die formalen Ziele der Steganographie sind Unentdeckbarkeit, die Maximierung der Datenkapazität und Robustheit gegen verschiedene Verarbeitungsschritte und Kompression (siehe Cheddad u.a. (2010)). Bei dem Entwurf eines steganographischen Systems müssen alle drei Ziele gegeneinander abgewogen werden: eine wesentliche Verstärkung eines dieser Aspekte beeinträchtigt typischerweise die Güte der anderen beiden.

Wir beschränken uns in diesem Kapitel auf die Übertragung von Bildern. Dies ist das derzeit wohl gebräuchlichste steganographische Szenario, da fast jeder Zugang zu einer digitalen Kamera hat und Bilder auch digital verbreiten kann (zum Beispiel per Email oder über Webplattformen). Bilder, die für den Austausch geheimer Nachrichten benutzt werden, nennt man Maske oder „Cover Channel“. Gleichzeitig bieten Bilder viel Raum, um zusätzliche Information einzubetten. Diese beiden Eigenschaften machen digitale Bilder als Cover Channel so populär.

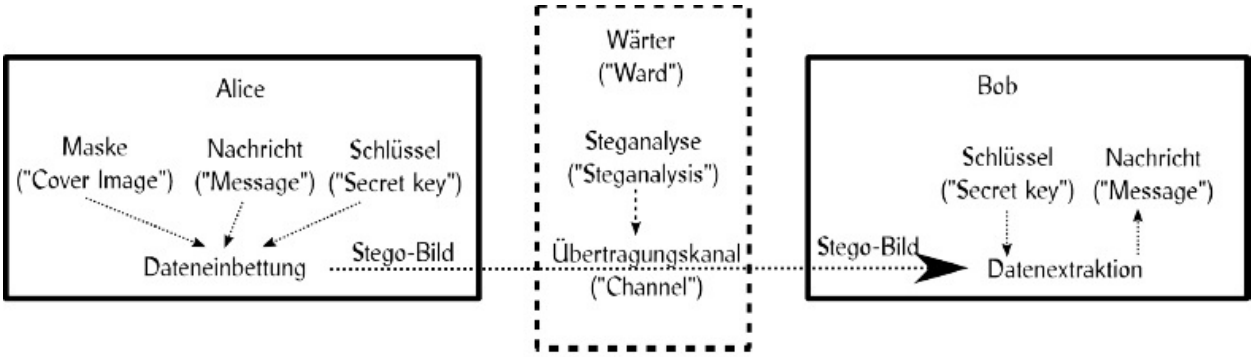


Abbildung 4.1: Prisoner’s Model (nach Li u. a. (2011)).

Das „Prisoner’s Model“

Formal wird zur Beschreibung eines steganographischen Szenarios häufig das „Gefangenenmodell“ („Prisoner′s Model“) herangezogen. Abbildung 4.1 (nach Li u. a. (2011)) zeigt das Kommunikationsmodell zwischen Alice („A“) und Bob („B“). Gemäß dem Modell sitzen Alice und Bob in Gefängniszellen, in denen sie nicht direkt überwacht werden. Der Austausch von Nachrichten zwischen den Zellen wird jedoch von dem Wärter kontrolliert. Alice wählt ein Bild als Maske. Hierin wird mit Hilfe eines geheimen Schlüssels eine Nachricht versteckt. Das hieraus erzeugte Bild wird auch „Stego-Bild“ genannt. Auch wenn die Tätigkeit in der Zelle nicht überwacht wird, muss die Informationsübertragung

unauffällig, effizient und robust sein gegen Eingriffe Dritter, wie z.B. Umkodierung des Bilds während der Übertragung.

Während das Stego-Bild in die Zelle von Bob übertragen wird, ist jederzeit davon auszugehen, dass der Wärter das Bild inspiziert. Techniken zur Erkennung von Steganographie werden „Steganalyse“ genannt. Je nach Szenario kann der Wächter aktiv oder passiv sein. Ein aktiver Wächter darf begrenzt in die Kommunikation eingreifen, beispielsweise Bilder umcodieren, oder ausgewählte Bilder löschen. Findet der Wärter nichts verdächtiges, und übersteht die eingebettete Nachricht eventuelle Eingriffe eines aktiven Wächters, kann Bob mit Hilfe seines geheimen Schlüssels die Nachricht lesen.

Least-Significant Bit-Einbettung

Wie kann eine geheime Nachricht in einem Bild versteckt werden? Wir betrachten exemplarisch den einfachsten Fall, die sogenannte Least-Significant Bit (LSB)-Einbettung. Hierfür ist es wichtig, den Aufbau und die Art der Speicherung von digitalen Bildern zu kennen.

Exkurs 5 (Pixel) ***Ein digitales Bild ist aus Punkten zusammengesetzt, sogenannten Pixeln. Ein Pixel ist üblicherweise eine Mischung aus einem roten, grünen und blauen Farbkanal. Jedem Farbkanal wird ein Wert zwischen 0 und 255 zugeordnet (Intensität genannt), die Farbe ergibt sich aus der Kombination der drei Kanäle. Beispielsweise bezeichnet*** (0, 0, 0) ***schwarz***, (255, 255, 255) ***weiß***, (255, 0, 0) ***ein sattes rot, und*** (128, 0, 128) ***ein lila (also halb rot, halb blau). In Graphikprogrammen wie dem Gimp (siehe Gimp (2012)) kann man sich für die Umsetzung von RGB-Tripeln in Farben ein Gefühl verschaffen, indem man verschiedene Werte in den Farbregler eingibt. Der Zahlenbereich von 0 bis 255 ist nicht zufällig gewählt: die*** $256 = 2^8$ ***möglichen Werte lassen sich exakt in acht Bit codieren, so dass pro Farbkanal ein Byte Speicher benutzt wird***.

Least-Significant Bit-Einbettungen verändern die Farbkanäle bitweise an der letzten Stelle, dem „am wenigsten wichtigen“ Bit. Das letzte Bit bestimmt, ob eine Zahl gerade oder ungerade ist. Beispielsweise unterscheiden sich 252 und 253, oder 254 und 255 genau in dem letzten Bit. Die Idee ist, dass für das Auge benachbarte Farben, zum Beispiel die Rottöne (255, 0, 0) und (254, 0, 0), kaum unterscheidbar sind. Dies kann genutzt werden, um die Stego-Nachricht

einzubetten. Einbetten und Auslesen ist dann sehr einfach: Durch Prüfung und Veränderung eines Farbwerts auf einen geraden oder ungeraden Wert kann jeweils ein Bit der Nachricht eingebettet werden. Die komplette Nachricht besteht aus den letzten Bits aller Pixel und aller Farbkanäle. Um die Nachricht statistisch etwas unauffälliger zu machen, wird sie vor der Einbettung oft verschlüsselt. Damit werden die 0- und 1-Werte der Nachricht ungefähr gleichverteilt.

Steganalyse und Steganographie: Ein Katz- und Maus-Spiel

Wie gut ist das vorgestellte LSB-Verfahren? Eine wichtige Kennzahl ist die Kapazität („Capacity“, oft auch „Payload“ genannt). Die Kapazität bezeichnet das Verhältnis von der Länge der geheimen Nachricht zu der Länge „unnützer“ Information in der Maske. In unserem Fall wird eines von acht Bit genutzt um die Nachricht einzubetten, die Kapazität ist also 1/8 = 0,125. Wird nicht jeder Farbkanal oder nicht jeder Pixel genutzt, verschlechtert sich die Kapazität entsprechend. Bei eigenen Experimenten lässt sich schnell feststellen, dass diese Einbettung nicht sonderlich robust ist: Die vorgestellte LSB-Einbettung kann beispielsweise nicht für JPEG-Bilder benutzt werden, da der JPEG-Algorithmus selbst die letzten Bits der Bildpixel verändert. Die LSB-Methode ist also darauf angewiesen, dass die Bilder verlustfrei gespeichert werden, beispielsweise in dem PNG-Format. Andere, hier nicht betrachtete Verfahren[5] beheben diese Schwäche: Da eine wichtige Komponente des JPEG-Verfahrens eine Frequenztransformation ist, kann die Nachricht ebenfalls im Frequenzraum eingebettet werden. Damit kann die Nachricht vor der Zerstörung durch den JPEG-Algorithmus geschützt werden.

Nach Diskussion von Kapazität und Robustheit betrachten wir nun die Sicherheit von LSB. Visuell ist die Methode sehr unauffällig. Aber mathematisch? Es ist davon auszugehen, dass der Wärter über technische Mittel verfügt, um das Bild gründlich zu analysieren. Durch die Einbettung der Nachricht wird das Bild leicht verändert. Da der Wärter nicht weiß, welche Nachricht eingebettet ist, kann er nicht direkt danach suchen. Er kann jedoch untersuchen, ob das übertragene Bild „unnatürlich“ ist — vorgeblich werden ja einfach nur Bilder ausgetauscht — und damit die geheime Kommunikation aufdecken.

Zur Bewertung der Natürlichkeit eines Bildes werden statistische Verteilungen benutzt (die sogenannte „Cover-Verteilung“). Beispielsweise kann

die Ausgangshypothese sein, dass in natürlichen Bildern gleich viele gerade wie ungerade Pixelintensitäten vorkommen. Wenn ein zu untersuchendes Bild nur aus geraden Pixelintensitäten bestehen, ist es verdächtig, und wird als potentielles Stego-Bild gemeldet. In der Praxis unterscheidet sich die Statistik des Stego-Bilds (auch „Stego-Verteilung“ genannt) jedoch oftmals schwächer von einer natürlichen Verteilung, und in subtileren Merkmalen. Für eine mathematisch fundierte Aussage zu der Ähnlichkeit von Cover-Verteilung und Stego-Verteilung wird typischerweise ein statistischer Test verwendet. Hierbei bildet die Verteilung eines natürlichen Bilds die sogenannte Null-Hypothese, also den Normalzustand. Die Statistik des fraglichen Bildes wird mit der Null-Hypothese getestet. Sollten die Verteilung des fraglichen Bilds zu weit von der Null-Hypothese entfernt sein, wird es als Stego-Bild markiert. Die Stärke der erlaubten Abweichung wird über ein sogenanntes Konfidenzintervall reguliert.

Ein gutes steganographisches Verfahren wird also versuchen, die Stego-Verteilung möglichst ähnlich zu der Cover-Verteilung zu wählen. Im Idealfall ändert die Einbettung der Nachricht die statistischen Eigenschaften des Bildes nicht. In einem solchen Fall ist die Einbettung unerkennbar. Wenn die Stego-Verteilung exakt gleich der Cover-Verteilung ist, spricht man von „perfekter Sicherheit“. Perfekte Sicherheit wird in der Praxis nicht erreicht, ist jedoch als theoretisches Konzept nützlich zur Gütebewertung eines steganographischen Verfahrens.

Zurück zur LSB-Einbettung. In diesem Fall gibt es zahlreiche steganalytische Verfahren, die LSB-Einbettungen aufdecken können (siehe Li u.a. (2011)). Ein effektiver statistischer Test ist der sogenannte χ^2–Test (sprich: „chi-Quadrat-Test“). Zur Vorbereitung des Tests wird ein Histogramm der Intensitäten pro Farbkanal aufgestellt. Durch die LSB-Einbettung wird eine Intensität zwischen $2\boldsymbol{n}$ und $2\boldsymbol{n} + 1$ variiert. Die Einbettung der Nachricht verändert das Histogramm also in den Zellenpaaren gerade Intensität/ungerade Intensität. Wenn die Vorkommen von 0 und 1 in der Nachricht in etwa gleich wahrscheinlich sind, treten im Histogramm benachbarte Intensitäten ebenfalls gleich häufig auf — in natürlichen Bildern ist dies nicht der Fall. Der statistische Test wird also folgendermaßen aufgebaut: Die Null-Hypothese ist die Annahme, dass Intensitäten in benachbarten Histogrammzellen unterschiedlich sind. Sollte das Gegenteil zu oft eintreten, muss man davon ausgehen, dass eine LSB-Einbettung vorliegt (siehe z.B. Fridrich (2000)). Die Bewertung, ab wann eine statistisch relevante Abweichung vorliegt wird von dem χ^2–Test übernommen. Einzelheiten zu dem χ^2-Test finden sich in der Statistik-Literatur (siehe z.B.

Papula (2006)), für uns genügt es an dieser Stelle zu wissen, dass bei Anwendung des Tests die (Un-)Ähnlichkeit von zwei Verteilungen quantifiziert wird.

Interessanterweise lässt sich mit einer leichten Veränderung der LSB-Einbettung die skizzierte Analyse abwehren. Anstatt Intensitäten zwischen $2\boldsymbol{n}$ und $2\boldsymbol{n} + 1$ zu variieren, kann ein Originalpixel durch Addition oder Subtraktion von 1 verändert werden, um zwischen geraden und ungeraden Werten umzuschalten. Die Variation ist nur minimal: Anstatt statisch eine 1 durch Umschalten von $2\boldsymbol{n}$ auf $2\boldsymbol{n} + 1$ zu erzeugen, kann nun auch $2\boldsymbol{n} - 1$ eine 1 darstellen. Die Einbettung der 0 wird analog gehandhabt. Damit sind die Histogrammzellen $2\boldsymbol{n}$ und $2\boldsymbol{n} + 1$ nicht fest gekoppelt, der $\boldsymbol{\chi}^2$–Test läuft ins Leere. Weitere Details zu LSB-Einbettungen und den vorgestellten Analysemethoden sind z.B. in Li u. a. (2011) enthalten.

Wir sehen, dass Steganographie und Steganalyse sich in einem steten Wettstreit befinden. Wird ein neues steganographisches Verfahren vorgestellt, beginnt die Suche nach einem statistischen Modell, um dieses Verfahren anzugreifen. Für die weiterführende Lektüre im Bereich Steganographie empfiehlt sich ein Blick in den Übersichtsartikel von Cheddad u. a. (2010).

4.1.2 Watermarking: Schutz geistigen Eigentums

Watermarking ist technisch nah verwandt mit Steganographie, allerdings sind die Ziele etwas unterschiedlich: Im Watermarking wird eine Markierung in ein Bild eingebettet, um den Urheber des Bildes identifizieren zu können. Im Unterschied zu den mechanischen Wasserzeichen, wie sie beispielsweise in Banknoten benutzt werden, soll die elektronische Markierung jedoch so unauffällig wie möglich sein, idealerweise komplett unsichtbar.

In der Anwendung gibt es zwei Varianten:

- ***Robuste*** Wasserzeichen sollen aus einem Bild nicht entfernbar sein ohne das Bild selbst zu zerstören. Damit kann der Urheber eines Bildes auch nach verschiedenen Bearbeitungsschritten seine Rechte an dem Bild beanspruchen.
- ***Fragile*** Wasserzeichen hingegen sind mit dem Ziel entworfen, dass sie mit dem Verändern des Bildes verschwinden. Kann in einem Bild ein fragiles Wasserzeichen nicht verifiziert werden, ist dies ein Indikator für eine Manipulation.

Wir betrachten jedoch im Folgenden nur robuste Wasserzeichen.

Exkurs 6 (alternative Trägermedien) ***Genau wie steganographische Nachrichten, sind auch digitale Wasserzeichen nicht auf Bilder beschränkt. Es gibt auch Wasserzeichenalgorithmen für Tondokumente, Videos oder gedruckte Texte. In vergangenen Zeiten waren beispielsweise Logarithmentafeln sehr teuer in der Anfertigung. Die Autoren dieser Bücher setzten absichtlich „Tippfehler“ in einigen Nachkommastellen, um Plagiatoren überführen zu können: Taucht eine andere Logarithmentafel auf dem Markt auf, kann der Autor des Originalwerks überprüfen, ob die fehlerhaften Einträge auch darin enthalten sind. Wenn ja, ist davon auszugehen, dass sie abgeschrieben wurden. Ähnliche Vorgehensweisen existieren auch im Bereich der Straßenkarten. Hier werden kleine Straßen in die Karte eingefügt, die gar nicht existieren. Taucht eine dieser Straßen in einer fremden Karte auf, so handelt es sich höchstwahrscheinlich um eine Kopie der Originalkarte***.

Allgemeiner Ansatz

Ein Wasserzeichen ist eine Bitfolge („Nachricht“), die über das Bild verteilt wird. Insofern ist die Wasserzeicheneinbettung sehr ähnlich zu einer steganographischen Einbettung: Farb- oder Grauwerte werden gezielt verändert, ohne den Bildinhalt auffällig zu beeinflussen. Im Detail gibt es jedoch wichtige Unterschiede zur Steganographie.

Abbildung 4.2 zeigt die generelle Vorgehensweise zur Einbettung eines digitalen Wasserzeichens. Die erste Überlegung ist, wo das Wasserzeichen in dem Bild eingebettet werden soll. Hierfür sind drei gegenläufige Randbedingungen zu beachten: Zum ersten soll der gesamte Bildbereich von dem Wasserzeichen überdeckt werden, um zu verhindern, dass lediglich ein Teil des Bildes missbräuchlich von Dritten verwendet wird. Zum zweiten soll das Wasserzeichen robust sein gegen Manipulationen. Zum dritten soll das Wasserzeichen visuell unauffällig sein. Hierfür wird das Wasserzeichen in sogenannten „perzeptionell geeigneten Regionen“ stärker eingebettet. Abhängig von dem gewählten Einbettungsverfahren können sich diese Regionen unterscheiden. Beispielsweise werden Farbveränderungen in homogenen Regionen oft vermieden, da sie dort sowohl stärker auffallen, als auch vergleichsweise leicht wegretuschiert werden können. Wird die Einbettung im

Ortsraum, d.h. direkt auf den Pixeln, vorgenommen, sind Objektkanten beliebte Einbettungsziele, da der Bildinhalt hier ohnehin stark variiert.

Das Wasserzeichen selbst besteht aus einer „Nachricht“, die beispielsweise den Eigentümer des Bildes codiert. Es ist oft sinnvoll, die Nachricht zusätzlich zu verschlüsseln, um ähnlich wie bei steganographischen Verfahren eine Gleichverteilung zwischen 0- und 1-Bits zu erhalten. Die Einbettung erfolgt mit einer robusten Methode, die es erlaubt, auch nach einer leichten Änderung des Bildinhalts (zum Beispiel durch Umcodierung des Bildes von dem PNG-Format in das JPG-Format) die Nachricht noch auszulesen.

Abbildung 4.3 (ebenfalls nach Pérez-González u. Hernández (1999)) zeigt das allgemeine Schema zum Finden und Auslesen eines Wasserzeichens. Auf dem (mutmaßlich mit einem Wasserzeichen versehenen) Bild werden analog zur Einbettung perzeptionell geeignete Regionen ermittelt. Dort wird das Wasserzeichen extrahiert. Oft wird angenommen, dass ein Original des Bildes ebenfalls verfügbar ist, um das Wasserzeichen sicherer zu lokalisieren.

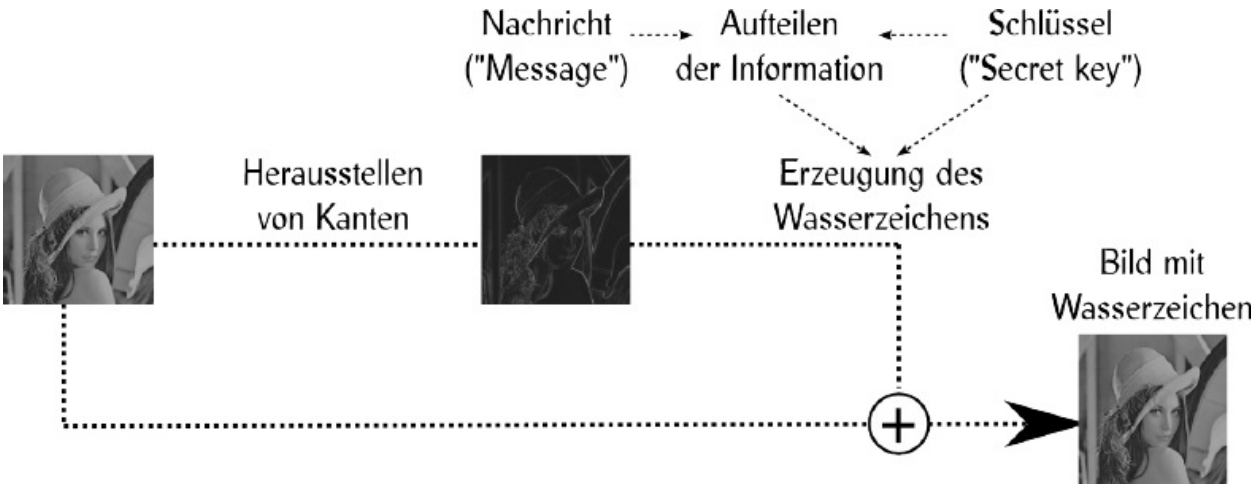


Abbildung 4.2: Einbettung eines digitalen Wasserzeichens (aus Pérez-González u. Hernández (1999)).

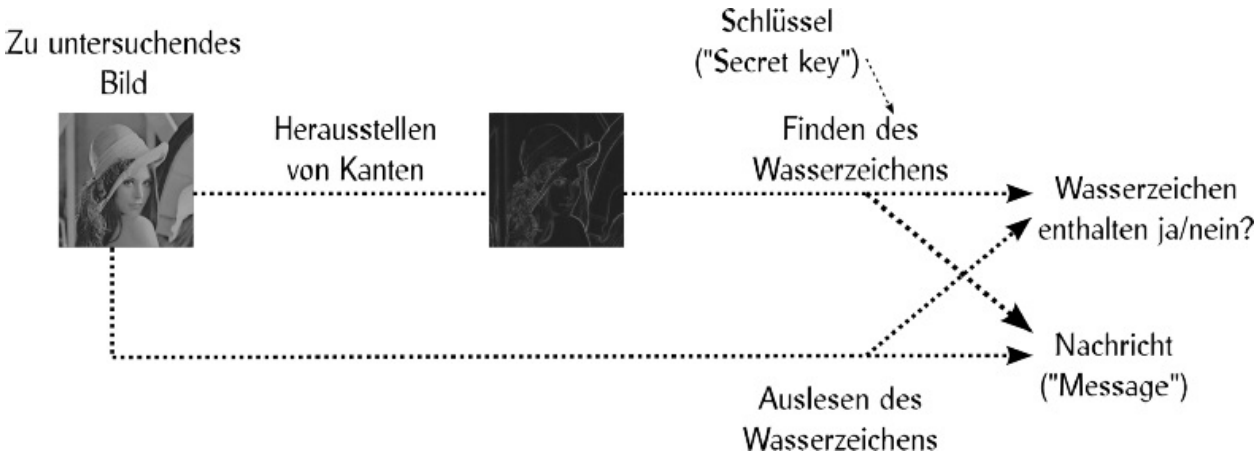


Abbildung 4.3: Auslesen eines digitalen Wasserzeichens (aus Pérez-González u. Hernández (1999)).

Das Vorhandensein eines Wasserzeichens wird über ein statistisches Verfahren ermittelt, beispielsweise durch die Berechnung der Korrelation oder der a posteriori-Wahrscheinlichkeit (siehe z.B. Cox u. a. (1997); Mairgiotis u. Galatsanos (2010)). Grundsätzlich ist es möglich, dass in das selbe Bild unterschiedliche Wasserzeichen eingebettet wurden. Genauso kann ein Bild an unterschiedliche Empfänger mit verschiedenen Wasserzeichen ausgeliefert werden. Eine bekannte Anwendung für letzteren Fall ist das am Ende des Abschnitts vorgestellte „Fingerprinting“.

Beispiel: Spread-Spectrum Wasserzeichen

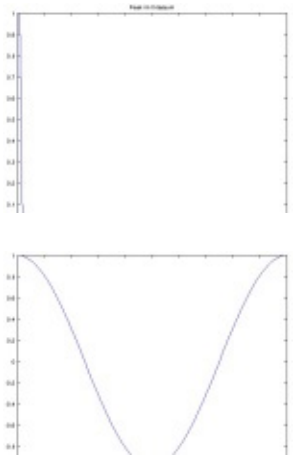
Ein robustes Wasserzeichen zeichnet sich dadurch aus, dass es schwierig ist, das Wasserzeichen zu entfernen ohne das Bild zu zerstören. Dies ist ein wesentlicher Unterschied zur Steganographie, in der Robustheit (beispielsweise gegen Umkodierung) lediglich wünschenswert ist. Wir haben gesehen, dass die in Abschnitt 4.1.1 vorgestellte LSB-Einbettung nicht robust ist gegen Umkodierung in das JPEG-Format. Damit ist sie in dieser Form als robustes Wasserzeichen ungeeignet.

Es gibt eine Vielzahl von Methoden zur Einbettung robuster Wasserzeichen. In diesem Abschnitt wird eine „Spread-Spectrum-Methode“ von Cox u. a. (1997) vorgestellt, die resistent ist gegen verschiedene Angriffe. Das Verfahren von Cox u. a. (1997) steht an dieser Stelle exemplarisch für die Gruppe von Verfahren, die auf einer Einbettung des Wasserzeichens in den Frequenzraum

beruhen. Alternativ ist es auch möglich, Wasserzeichen direkt im Ortsraum einzubetten.

Exkurs 7 (Frequenzräume) ***In der Signalverarbeitung werden Orts- und Frequenzraum unterschieden. Der Ortsraum bezeichnet die Messungen über die Zeit oder eine räumliche Ausdehnung, zum Beispiel die Amplitude eines Sprachsignals, oder die Pixel eines Bildes. Intuitiv ist der Ortsraum die „normale“ Darstellung eines Bildes, wie wir sie z.B. aus Bildverarbeitungsprogrammen kennen. Der Frequenzraum ist eine äquivalente, alternative Darstellung. Am bekanntesten ist die Fourier-Transformation, die mit Sinus- und Cosinustermen arbeitet. Eine weitere häufig genutzte Methode ist die diskrete Cosinus-Transformation (DCT), die lediglich mit dem Cosinus arbeitet. Wir betrachten Beispiele für die DCT in einer und zwei Dimensionen in Abbildung 7 auf der nächsten Seite***.

Die obere Zeile von Abbildung 7 auf der nächsten Seite zeigt den Ortsraum, von links nach rechts eine einzelne Spitze in 1D, eine Welle in 1D, eine 2D-Spitze und eine 2D-Welle. Die zweiten Zeile enthält die dazugehörigen Frequenzräume. Die Höhe oder Intensität eines Punkts im Diagramm bezeichnet die Stärke der jeweiligen Frequenz. Vom Orts- zum Frequenzbereich werden Wellen zu Spitzen, und umgekehrt. Im 2D-Fall ist der Koordinatenursprung links oben. Reale Bilder enthalten wesentlich komplexere, unregelmäßigere Strukturen. Die Frequenzdarstellung ist dann eine Überlagerung einfacherer Bildelemente.



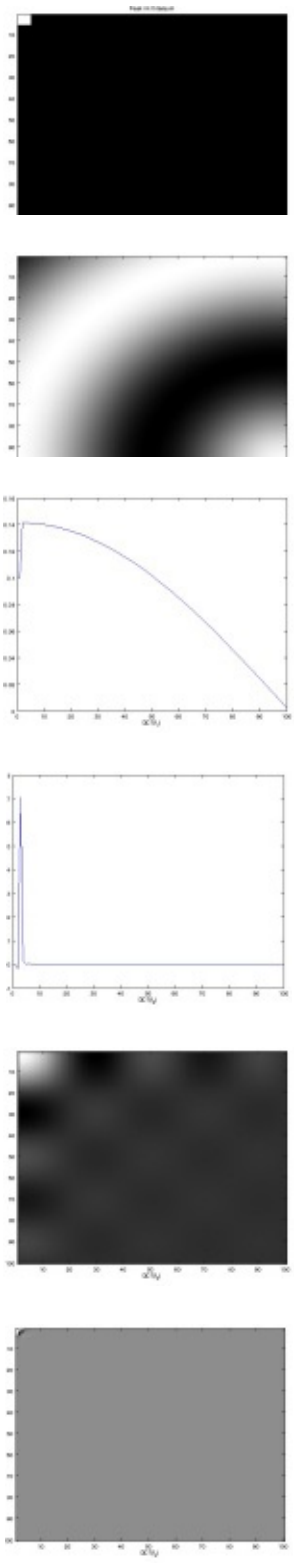


Abbildung 4.4: Ortsraum und Frequenzraum.

Um die Einbettung in den Frequenzraum zu verstehen, ist Kenntnis über die Repräsentation eines Bildes im Frequenzraum nötig. Die Frequenzdarstellung entsteht wir folgt: Die Intensitäten eines Bildes werden in eine Summe horizontaler und vertikaler Schwingungen unterschiedlicher Frequenzen umgerechnet. Die Frequenzen können wiederum als Bild gezeichnet werden, hellere Punkte zeigen stärkere Frequenzen. Typischerweise werden tiefe Frequenzen in der Mitte und hohe Frequenzen am Rand angeordnet. Wegen Symmetrien im Frequenzraum wird (wie in der obigen Abbildung) manchmal auch nur der rechte untere Quadrant visualisiert.

Exkurs 8 (Frequenztransformation) ***Die Frequenztransformation ist eine verlustfreie, umkehrbare Transformation. In einigen Ingenieursdisziplinen, wie zum Beispiel Signalverarbeitung, ist sie eines der wichtigsten mathematischen Werkzeuge überhaupt. In der Bildverarbeitung werden Frequenztransformationen oft zweidimensional (in x-und y-Richtung) angewandt. Beliebte Algorithmen sind die diskrete Fourier-Transformation und die diskrete Cosinus-Transformation***.

Cox u.a. (1997) benutzen Cosinus-Schwingungen (die sogenannte „Diskrete Cosinus-Transformation“) für die Wasserzeicheneinbettung. Abbildung 4.5 zeigt auf der linken Seite das Testbild von Cox u.a. (1997). In der Mitte ist eine Visualisierung des Frequenzraums angegeben. Das Wasserzeichen wird in die stärksten Frequenzen eingebettet, d.h. im zentralen Teil des visualisierten Spektrums (rechtes Bild). Diese Einbettung ist robust, da die stärksten Frequenzen die meiste Bildinformation tragen, und dementsprechend nicht einfach entfernt oder grob verändert werden können ohne das Bild wesentlich zu beschädigen.

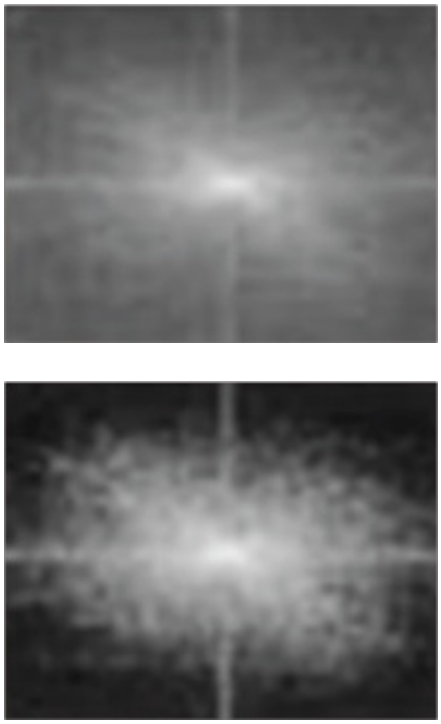


Abbildung 4.5: Einbettung eines Wasserzeichens im Frequenzraum, nach Cox u. a. (1997).

Die eigentliche Einbettung ist mathematisch nicht schwierig zu bewerkstelligen. Sei das Wasserzeichen $\boldsymbol{n}$ Bits lang, $\boldsymbol{v_i}$ eine der $\boldsymbol{n}$ stärksten Frequenzen im Bild, und $\boldsymbol{x_i}$ das $\boldsymbol{i}$-te Bit im Wasserzeichen (mit Wert 0 oder 1). Dann wird das Wasserzeichen $\boldsymbol{v_i}$ eingebettet durch die Berechnung von

$$v_i' = v_i \cdot (1 + \alpha x_i) \ . \tag{4.1}$$

Hierbei ist $\boldsymbol{\alpha}$ ein Skalierungsparameter, der die Stärke des Wasserzeichens bestimmt. Größere Werte für $\boldsymbol{\alpha}$ führen zu größerer Robustheit, aber auch zu stärkerer Veränderung des Bildinhalts. Cox u.a. (1997) benutzen den Wert $\boldsymbol{\alpha}$ = 0.1. Durch die Umkehrtransformation des Frequenzraums in den Ortsraums erhält man wieder das Bild, das nun das Wasserzeichen enthält.

Die Prüfung, ob ein Wasserzeichen eingebettet ist, funktioniert nun folgendermaßen: Nehmen wir an, wir veröffentlichen ein Bild $\boldsymbol{I}$ mit Wasserzeichen $\boldsymbol{X}$ und finden nun im Internet ein fast identisches Bild $\boldsymbol{I'}$. Um zu prüfen, ob $\boldsymbol{I'}$ unser Wasserzeichen enthält, werden $\boldsymbol{I}$ und $\boldsymbol{I'}$ in den Frequenzraum transformiert. Die $\boldsymbol{n}$ höchsten Frequenzen von $\boldsymbol{I}$ sollten als auch in $\boldsymbol{I'}$ das

Wasserzeichen enthalten. Wir subtrahieren die Frequenzdarstellungen des gefundenen Bilds ***I'*** und des Originalbilds ***I***, und erhalten so das aus ***I'*** extrahierte Wasserzeichen $\boldsymbol{X}^*$. Der Grad der Übereinstimmung zwischen ***X*** und $\boldsymbol{X}^*$ kann nun beispielsweise über den Korrelationskoeffizienten berechnet werden. Cox u. a. (1997) bevorzugen als Ähnlichkeitsmaß jedoch

$$\mathrm{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}} \quad . \tag{4.2}$$

Die Autoren demonstrieren, dass diese Art der Wasserzeicheneinbettung und -prüfung unter anderem resistent ist gegen JPEG-Umkodierung, Dithering (d.h. Rasterung von Grauwerten in schwarze und weiße Bildpunkte), teilweise Ersetzung des Bildinhalts, Einbettung weiterer Wasserzeichen, sowie Ausdrucken und erneutes Digitalisieren (Scannen) des Bilds.

Anwendung: Fingerprinting

Als „Fingerprinting“ bezeichnet man die Einbettung identifizierender Merkmale in Daten. Digitale Wasserzeichen eignen sich hierfür hervorragend, so dass dieses Anwendungsfeld für digitale Wasserzeichen sich in den letzten Jahren zu einem eigenen Forschungszweig entwickelt hat.

Ein typisches Anwendungsszenario wird im folgenden betrachtet. Nehmen wir einen Inhalteanbieter im Internet, beispielsweise eine Web-Plattform, auf der gegen Bezahlung Filme angesehen werden können. Fingerprinting ist eine Schutzmaßnahme gegen die illegale Weiterverbreitung von Kopien dieser Filme. Hier genügen einfache Wasserzeichen nicht, weil von vorneherein klar ist, dass die kursierende Kopie des Films illegal angefertigt wurde. Interessanter ist für das Unternehmen, ***wer*** den Film in Umlauf gebracht hat. Wenn der Film ursprünglich nur an registrierte, zahlende Kunden ausgeliefert wurde, ist der Kreis der Verdächtigen stark eingeschränkt.

Die Idee ist nun, in jede Kopie des Films ein eigenes Wasserzeichen einzubetten, um so den Täter zu identifizieren. Technisch ist das kein Problem, allerdings kann dieser (naive) Ansatz angegriffen werden, wenn eine Gruppe sogenannter „Verschwörern“ zusammenarbeitet. Wenn mehrere Personen legale

Kopien des selben Videos besitzen, dann unterscheiden sich die Videodateien lediglich in den Wasserzeichen. Verschwörer könnten also die Stellen, in denen sich ihre Videos unterscheiden, gezielt abändern oder entfernen. Damit kann das Wasserzeichen stark beschädigt oder entfernt werden, so dass eine Rekonstruktion des Verbreitungswegs nicht mehr möglich ist.

Fingerprinting versucht diese Schwachstelle zu beheben. Hierbei wird das eingebettete Wasserzeichens mit einem speziellen Codegenerator konstruiert. Ein Wasserzeichen ist ein „Codewort“, also eine Zeichenfolge mit bestimmten mathematischen Eigenschaften. Es wird ein Code benutzt, bei dem zwei Codeworte sich jeweils nur in einigen Stellen unterscheiden. Wenn zwei Verschwörer zusammenarbeiten, können sie durch Vergleich Ihrer Kopien also nur einen Teil des Wasserzeichens auslöschen, der andere Teil bleibt erhalten. Die Pointe des Verfahrens ist, dass aus dem nicht ausgelöschten Teil Rückschlüsse auf die Identität der Verschwörer gezogen werden kann. Mit anderen Worten ist jedes Paar von Codeworten eindeutig in den Stellen, an denen die beiden Codeworte sich unterscheiden.

Eine Herausforderung bei der Konstruktion des Codes ist die effiziente Prüfbarkeit der extrahierten, möglicherweise teilweise beschädigten Codeworte. Dies wurde durch die Arbeit von Tardos praktikabel (siehe Tardos (2003)). Für die Konstruktion des Codes muss eine Maximalzahl von Verschwörern angenommen werden. Diese Zahl liegt typischerweise zwischen 2 und 5.

4.1.3 Blinde Bildforensik: Authentizität eines Bildes oder Videos

Fragile und robuste Wasserzeichen können Authentizität und Herkunft eines Bildes oder Videos belegen. Sobald ein Wasserzeichen eingebettet wurde, ist der Inhalt relativ zuverlässig geschützt. Diese Einbettung wird in der Praxis jedoch nicht flächendeckend durchgeführt. Bei dem breiten Angebot digitaler Kameras und Programmen zur Bearbeitung digitaler Bilder oder Videos kann nicht vorausgesetzt werden, dass jedes Dokument ein Wasserzeichen enthält. Die „blinde Bildforensik“ füllt diese Lücke: Angenommen, wir haben Bilder aus (möglicherweise unbekannter) Quelle und ohne eingebettete Sicherheitsmechanismen. Welche Aussagen können wir in diesem Fall über die Authentizität oder das Aufnahmegerät treffen? Dieses Thema macht den

verbleibenden Hauptteil des Kapitels aus. In dem folgenden Abschnitt werden die grundlegenden Ansatzpunkte vorgestellt. In Abschnitt 4.2 betrachten wir konkrete Methoden, um die Aufnahmekamera digitaler Bilder zu identifizieren. In Abschnitt 4.3 werden verschiedene Ansätze diskutiert, um Manipulationen in Bildern aufzudecken.

Geräte oder Inhalte?

Zwei Fragestellungen dominieren die Forschung in der blinden Bildforensik:

1. die Identifikation von Aufnahmegeräten und
2. die Authentifizierung von Inhalten.

Bei der Identifikation von Aufnahmegeräten setzt sich die klassische Forensik im Multimedia-Bereich fort: Beispielsweise hinterlässt jede Feuerwaffe charakteristische Signaturen auf dem Geschoss. Sobald ein Ballistiker in den Besitz der Feuerwaffe und des fraglichen Projektils kommt, lässt sich mit gezielten Experimenten ermitteln, ob das Projektil aus der vorliegenden Waffe abgeschossen wurde. In der Bildforensik wird ein ganz ähnlicher Ansatz verfolgt: Kein Kamerahersteller baut exakt die gleichen Kameras wie sein Konkurrent, so dass sich mit statistischen Methoden das Modell ermitteln lässt. Bei genauerer Untersuchung stellt sich sogar heraus, dass selbst Kameras der ***selben*** Modellreihe nicht exakt gleich sind. Hiermit kann eine eindeutige Beziehung zwischen einem Bild und dem Aufnahmegerät hergestellt werden.

Bei der Authentifizierung von Inhalten ist der Einsatz anderer Techniken sinnvoll. Selbst wenn ein Bild aus einer bestimmten Kamera stammt, so ist es doch möglich, dass die Bildaussage durch einen gezielten Eingriffe verändert wurde. Eine Herausforderung für den forensischen Gutachter ist hierbei die Abgrenzung „erlaubter“ Eingriffe von „unerlaubten“ Eingriffen. Ein erlaubter Eingriff ist jede Maßnahme, mit der die Bildqualität verbessert wird, die Bildaussage jedoch unverändert bleibt. Typische Beispiele sind die Anpassung von Helligkeit und Kontrast, oder die Anpassung des so genannten Gamma-Faktors (also ein nichtlinearer Helligkeitsabgleich). Unerlaubte Eingriffe sind jegliche Maßnahme, die die Bildaussage verfälscht. An der Grenze zwischen Bildverbesserung und Bildfälschung gibt es einen Graubereich: Eignet sich ein Eingriff dazu, die Bildaussage zu verändern oder nicht? Es ist nicht möglich, dies direkt zu prüfen. Als Ingenieursdisziplin kann die Bildforensik lediglich technische Eigenschaften eines Bildes aufdecken, die auf eine Veränderung der

Bildaussage hinweisen. Die Bewertung der Untersuchungsergebnisse obliegt typischerweise einem Gutachter.

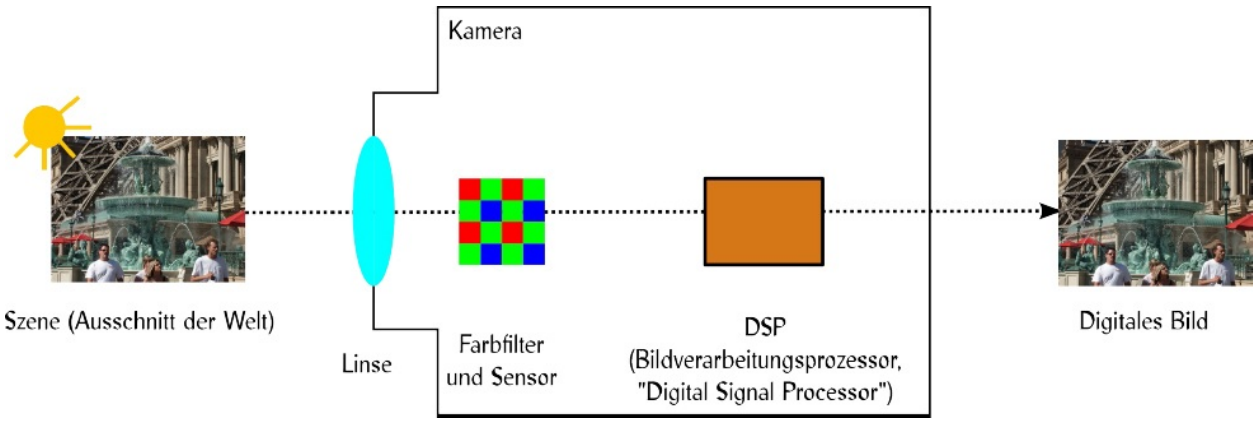


Abbildung 4.6: Bildentstehung von der aufgenommenen Szene bis zum digitalen Bild.

Ansatzpunkte für blinde forensische Methoden

Abbildung 4.6 zeigt die Abfolge der Bildentstehung von der aufgenommenen Szene bis zum digitalen Bild. Jeder Zwischenschritt kann forensisch ausgenutzt werden, um Herkunft und Authentizität des Bildes zu prüfen.

Zu Beginn des Aufnahmeprozesses steht die Szene. Sofern es sich um eine reale Szene handelt, müssen Lichteinfall, Schattenwurf und perspektivische Projektion den physikalischen Gesetzen gehorchen. Die Linse bricht das einfallende Licht nicht perfekt, so dass Farbverzeichnungen (chromatische Aberration) als Artefakte in dem digitalen Bild entstehen. Für die Aufnahme eines Farbbildes ist typischerweise ein Farbfilter (meist ein sogenanntes Bayer-Muster) vorgeschaltet. Die Farbfilter maskieren jede Sensorzelle, so dass entweder rote, grüne oder blaue Intensitäten aufgenommen werden. Die Sensorzellen weisen darüber hinaus fertigungsbedingt leicht unterschiedliche Sensitivitäten auf, so dass beispielsweise eine homogene Fläche in dem digitalen Bild nicht absolut homogen abgebildet ist. Anschließend wird das Bild in dem Digitalen Signalprozessor (DSP, „Digital Signal Processor“) aufbereitet. Aus den roten, grünen und blauen Bildpunkten wird mittels Interpolation ein durchgängig koloriertes Bild erstellt. Sehr hohe Intensitäten werden abgeschnitten („Clipping“), der Helligkeitsverlauf wird mittels Gamma-Abgleich nichtlinear an die menschliche Wahrnehmung angepasst. Kanten werden eventuell geschärft, und es wird ein Weißabgleich, also eine

Kompensation der Lichtfarbe, durchgeführt. Schließlich wird das Bild in einem Bildformat kodiert, beispielsweise in dem JPEG-Format. Damit liegt das digitale Bild vor.

Jeder Schritt in der Kamera fügt Abbildungsfehler („Artefakte“) in das digitale Bild ein. Viele Verfahren zur Manipulationserkennung zielen auf die Erkennung von Inkonsistenzen in diesen Artefakten ab. Eine weitere Möglichkeit ist, direkt auf dem fertigen Bild nach Fälschungsspuren zu suchen, beispielsweise nach geklonten Regionen.

Eine häufige Annahme bei der Entwicklung bildforensischer Verfahren ist, dass ein Fälscher lediglich nach künstlerischen Gesichtspunkten vorgeht. Das heißt, dass durch Betrachtung des Bilds die Fälschung nicht erkennbar ist. Aus diesem Grund zielen die meisten bildforensischen Methoden darauf ab, unsichtbare Spuren zu finden, deren Erzeugung ein Fälscher ohne technische Ausbildung vermutlich kaum vermeiden kann.

4.2 Identifikation des Aufnahmegeräts

Die Verknüpfung zwischen Aufnahme und Aufnahmegerät setzt die klassische Forensik im digitalen Bereich fort. Die Grenzen zur Manipulationserkennung sind manchmal fließend: Wenn nachweislich nur ein ***Teil*** des Bildes aus der gesuchten Kamera kommt, ist dies ebenfalls ein Hinweis auf eine Bildmanipulation. Wir betrachten exemplarisch drei Methoden, um die generelle Vorgehensweise zur Geräteidentifikation zu demonstrieren. Hierbei nimmt das Verfahren zur Extraktion des Sensorrauschens eine Sonderrolle ein. Die Forschungsresultate der letzten Jahre legen nahe, dass dies der momentan erfolgversprechendste Ansatz zur Kameraidentifikation ist. Die beiden anderen Verfahren wurden für die Unterscheidung von Kameramodellen vorgestellt. Obwohl die Modellunterscheidung weniger Aussagekraft hat als die direkte Zuordnung von Kamera zu Bild, sind Szenarien denkbar, in denen derartige Methoden nützlich sind.

4.2.1 Identifikation des Kameramodells

Wir betrachten zuerst die Zuordnung eines Bildes zu einem Kameratyp, also einem Hersteller oder einer Baureihe. Die erfolgreiche Nutzung der chromatischen Aberration hängt maßgeblich von der Qualität der Kameralinse

ab, während Bildqualitätsmerkmale auf Eigenschaften des digitalen Signalprozessors abzielen (vergleiche Abbildung 4.6).

Laterale Chromatische Aberration

Laterale chromatische Aberration ist eine Farbverzeichnung der Linse. Abbildung 4.7 zeigt schematisch, wie chromatische Aberration entsteht. Eine zentrale Rolle spielen die unterschiedlichen Wellenlängen des Lichts, die vom Menschen als unterschiedliche Farben wahrgenommen werden. Unterschiedliche Wellenlängen werden jedoch von einer Linse nicht gleich stark gebrochen. Abhängig davon, wie die Linse geschliffen ist, kann es also sein, dass beispielsweise der Inhalt des blauen Farbkanal im Vergleich zum roten Farbkanal leicht zum Bildrand hin gestreckt ist. Bei den Linsenherstellern ist dies ein bekanntes Problem: Es ist unmöglich, eine Linse ohne chromatische Aberration zu konstruieren. Durch eine geschickt gewählte Linsenform und moderne Fertigungsmethoden kann chromatische Aberration jedoch substantiell reduziert werden. Der Grad der Unterdrückung von chromatischer Aberration ist damit ein Qualitätsmerkmal von Kameralinsen. Im allgemeinen sind Farbverzeichnungen bei teureren Linsen schwächer als bei preiswerteren Linsen.

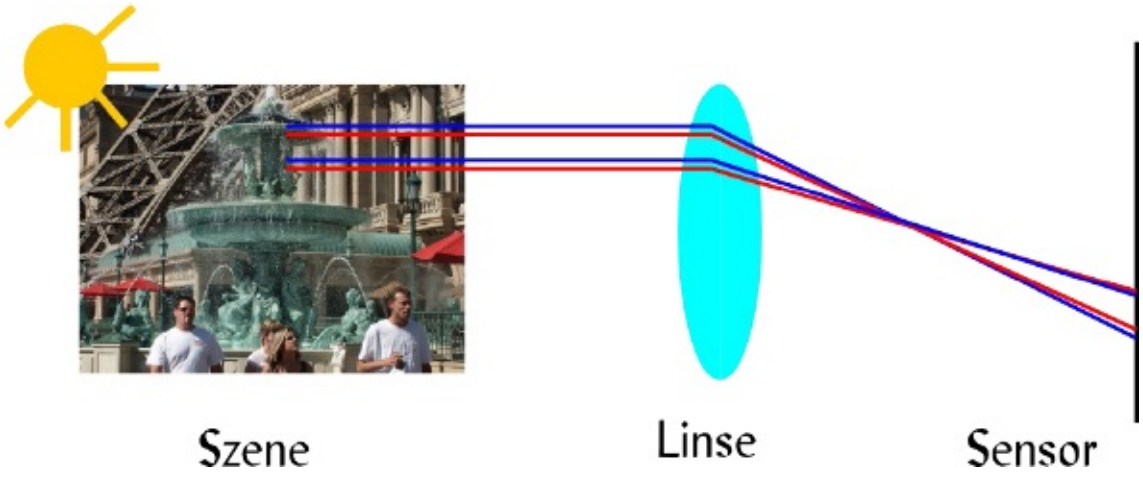


Abbildung 4.7: Schematische Darstellung von lateraler chromatischer Aberration. Lichtstrahlen unterschiedlicher Wellenlänge, die im selben Punkt auf der Linse auftreffen, werden leicht unterschiedlich gebrochen.

Johnson u. Farid (2006) schlugen 2006 ein lineares Modell vor, um die Farbverzeichnung durch laterale chromatische Aberration zu beschreiben und forensisch auszunutzen. Die Autoren nehmen an, dass die Farbverzeichnung als

lineare Streckung oder Stauchung der Farbkanäle modelliert werden kann: Sei $\vec{x}_s$ das Streckungszentrum im Bild, $\vec{x}_r$ die Position auf dem Sensor, an der die roten Lichtwellen auftreffen, und $\vec{x}_b$ die Position für blaue Lichtwellen. Dann verhalten sich die Farbkanäle zueinander gemäß

$$\vec{x}_b = \beta \cdot (\vec{x}_r - \vec{x}_s) + \vec{x}_s \quad , \tag{4.3}$$

wobei $\boldsymbol{\beta}$ der Streckungsfaktor zwischen den Farbkanälen ist. Der Wert $\boldsymbol{\beta}$ ist typischerweise sehr klein, so dass mit dem Auge nur am Rand des Bildes ein Farbversatz beobachtbar ist. Dennoch kann diese Streckung auch in anderen Bildregionen detektiert werden: Wenn der eine Farbkanal genau um den Faktor $\boldsymbol{\beta}$ gestreckt wird, passt er theoretisch besser auf den anderen als vorher. Die Passgenauigkeit zweier Farbkanäle kann mit speziellen Metriken gemessen werden, beispielsweise mit der Korrelation oder der Mutual Information.

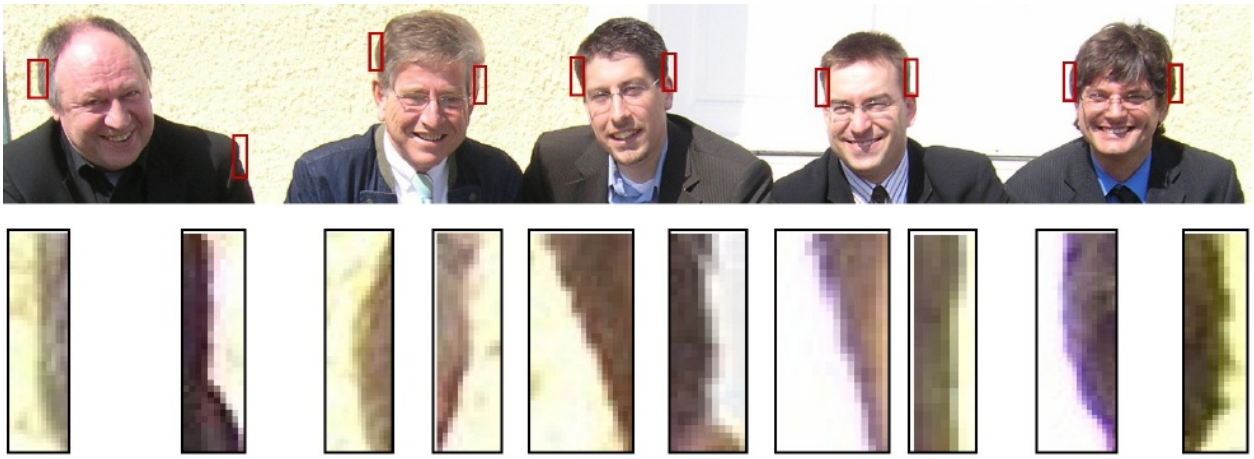
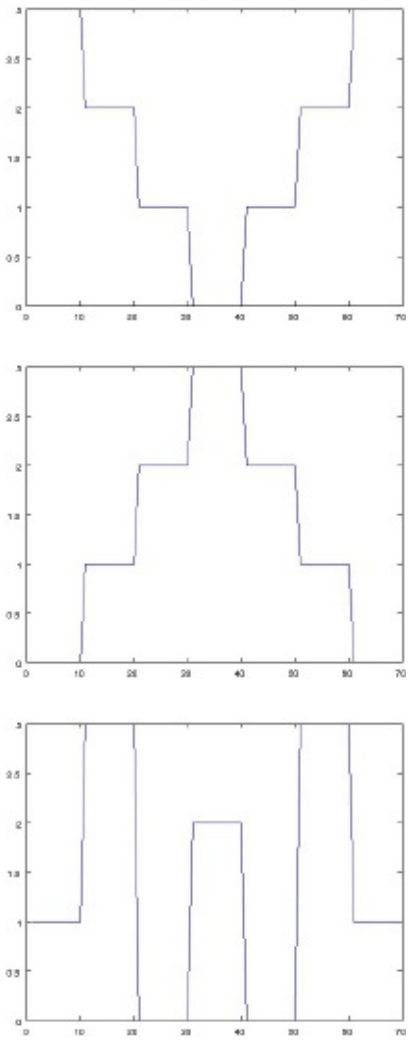


Abbildung 4.8: Beispiel für stark ausgeprägte chromatische Aberration: Die rot ausgewählten Bereiche des oberen Bilds sind unten vergrößert dargestellt. Durch die chromatische Aberration erscheinen an den inneren Objektkanten rötliche Schlieren.

Exkurs 9 (Korrelation und Mutual Information) ***Korrelation und Mutual Information sind zwei statistische Ähnlichkeitsmaße. Je stärker die Korrelation ausschlägt (positiv oder negativ), desto stärker sind zwei Variablen (oder Bilder, als Folge von Intensitäten) linear abhängig. Eine Korrelation von*** 0 ***zeigt an, dass keine solche Abhängigkeit identifiziert werden kann. Die Mutual Information ist etwas allgemeiner, mit ihr können auch nichtlineare Abhängigkeiten erfasst werden. Dazu folgendes Beispiel: Wir definieren drei Vektoren x, y, z, deren Werte von links nach rechts geplottet sind:***



corr(x, y) = −1, ***d.h. x und y sind linear abhängig (in diesem Fall ist nur das Vorzeichen unterschiedlich). Hingegen ist*** corr(x, z) = 0.1, ***d.h. es entsteht ein großer Fehler, wenn x mit einer linearen Abbildung z approximieren soll. Wir sagen, x und z sind kaum korreliert***.

Es ist jedoch nicht der Fall, dass x keine Information über z enthält: da jeder Wert von x eineindeutig auf z abgebildet wird, kann aus Kenntnis von x und dieser Abbildung z exakt rekonstruiert werden. Dies zeigt die Mutual Information: MutInf(x, z) = 1.95, ***bei einem Wertebereich von*** 0 (***keine Mutual Information) bis*** 2 (***vollständige Mutual Information***).

Dieses Gütemaß ermöglicht eine erschöpfende Suche über die Parameter x_s und β. Die Farbverzeichnung wird mit dem besten Parametersatz dargestellt.

Johnson und Farid schlagen nun vor, den besten Parametersatz zur Erkennung von Bildmanipulation zu nutzen, indem Regionen identifiziert werden, die nicht

in dieses Modell passen. Oft empfiehlt es sich, statt des roten und blauen den roten und grünen Farbkanal zu vergleichen, oder alternativ den blauen und grünen Farbkanal. Dies kommt von dem Umstand, dass die Linsen in der Fertigung oftmals so gekrümmt werden, dass der Farbverzeichnung zwischen rot und blau explizit entgegengearbeitet wird. In der Praxis ist die Methode dennoch schwierig anwendbar, wie zum Beispiel Gloe u. a. (2010) anmerken: Bei qualitativ höherwertigen Linsen ist das lineare Verzeichnungsmodell aus Gleichung 4.3 eine zu starke Vereinfachung, so dass die Methode im allgemeinen relativ unzuverlässig ist.

Eine andere Verwendung für Johnsons und Farids Idee schlugen jedoch Van u. a. (2007) vor: Anstatt nach Fälschungen zu suchen, können die Parameter $\boldsymbol{x_s}$ und $\boldsymbol{\beta}$ benutzt werden, um Kameras von Mobiltelefonen zu identifizieren. Bei solchen miniaturisierten, unter starkem Kostendruck gefertigten Linsen tritt die laterale chromatische Aberration relativ stark auf. Gleichzeitig ist die Unterscheidung von Kameramodellen eine weitere Vereinfachung, es entfällt nämlich die Suche nach inkonsistenten Regionen. In einer kleinen Studie konnten Van u. a. (2007) zwischen 86% und 96% der Bilder jeweils einem von drei Kameramodellen korrekt zuordnen.

Klassifikation von Bildqualitätsmerkmalen

Ein anderer Weg Kameramodelle zu unterscheiden wurde von Kharrazi u. a. (2004) vorgeschlagen. Anstatt nach physikalisch motivierten Merkmalen zu suchen, können in dem Ausgabebild die Besonderheiten des jeweiligen digitalen Signalprozessors identifiziert werden. Beispielsweise unterscheiden sich zwei Kameras in der Skalierung der Intensitäten, der Korrelationen zwischen zwei Farbkanälen, dem Spektrum der Farben, der Stärke der Kanten und dem Rauschniveau. Kamerapixel können zum Beispiel durch eine Glättungsoperation räumlich korreliert sein. Genauso kann es sein, dass die Farbkanäle — beispielsweise durch Weißabgleich oder durch Gamma — gespreizt wurden. Die Nachbearbeitung der Kanten dient einer lebendigeren Wiedergabe des Bildes, und wird häufig als Nebenprodukt der Interpolation der vorgeschalteten Bayer-Farbfilter mit durchgeführt.

Diese Eigenschaften explizit zu modellieren, wie im Beispiel der chromatischen Aberration, ist sehr schwierig. Kharrazi u. a. (2004) schlagen daher vor, heuristische Merkmale zur Bewertung der Bildqualität heranzuziehen,

und so indirekt auf die Kameraeigenschaften zu schließen. Die Autoren berechnen 34 Merkmale, wie zum Beispiel das quadrierte Verhältnis der grünen Intensitäten zu rot und blau, den Massenschwerpunkt benachbarter Intensitäten, und die Korrelation zwischen benachbarten Farbwerten. Jede dieser Eigenschaften wird in einen 34-elementigen Vektor eingetragen. Um nun zwei Kameramodelle zu unterscheiden, benutzen die Autoren eine „Support Vector Machine“. Dies ist ein Algorithmus aus dem maschinellen Lernen. Während einer Trainingsphase wird für je zwei Kameramodelle eine prototypische Kombination der Merkmale gesucht, für die es möglich ist, die Modelle zu unterscheiden. Mit dieser Voreinstellung können dann andere Bilder klassifiziert werden. Die Leistung des Klassifikators hängt dabei direkt mit der Eignung der Trainingsdaten ab. Die Autoren empfehlen, für das Training mit allen beteiligten Kameramodellen jeweils die selben Motive aufzunehmen, damit die Unterschiede in den resultierenden Bildern deutlicher hervortreten. Damit erreichten die Autoren eine durchschnittliche korrekte Klassifikationsrate von mehr als 98%.

Ein genereller Nachteil von Verfahren die auf heuristischen Merkmalen arbeiten ist die fehlende Transparenz von Ursache und Wirkung: Ein forensischer Gutachter hat kaum Möglichkeiten, das Resultat der Support Vector Machine selbst zu überprüfen, oder sich bezüglich der Fehlerwahrscheinlichkeit ein eigenes Bild zu machen. Obwohl mit der experimentellen Fehlerrate von 98% das Verfahren häufig richtig liegt, kann ein Einzelfallresultat mit dieser Methode kaum weiter untersucht werden.

4.2.2 Sensorrauschen zur Identifikation der Kamera

Das bis dato stärkste Verfahren zur Kameraidentifikation stammt von Lukáš u. a. (2006). Es basiert auf kleinen Fertigungsunterschieden in den Kamerasensoren: Die Lichtempfindlichkeit jedes Pixels ist nicht gleich und schwankt in einem (sehr kleinen) Wertebereich. Diese Varianz wird als Rauschen bezeichnet und ist der Ansatzpunkt für die Methode. Das Bildrauschen läßt sich aufschlüsseln in statisches Rauschen („Fixed Pattern Noise“) und belichtungsabhängiges Rauschen („Photo Response Non-Uniformity“, PRNU). Lukáš u.a. (2006) schlagen die Nutzung von PRNU für die Kameraidentifikation vor. Die Eigenschaft des Photo Response Non-Uniformity ist

1. charakteristisch für einen Sensor,

2. weitgehend unabhängig von dem aufgenommenen Motiv,
3. und weitgehend unabhängig vom Alter der Kamera und der Temperatur des Sensors.

Formal ausgedrückt nehmen die Autoren das folgende Rauschmodell für einen Pixel an Position (***i, j***) an:

$$y_{ij} = f_{ij} \cdot (x_{ij} + \eta_{ij}) + c_{ij} + \nu_{ij} \quad , \tag{4.4}$$

wobei die einzelnen Elemente der Gleichung die folgende Bedeutung haben:

- $\boldsymbol{y_{ij}}$ ist die beobachtete Pixelintensität.
- $\boldsymbol{x_{ij}}$ ist die ideale, rauschfreie Pixelintensität.
- $\boldsymbol{f_{ij}}$ ist das PRNU-Rauschen.
- $\boldsymbol{\eta_{ij}}$ bezeichnet temperaturunabhängiges Rauschen aufgrund diskreter Elektronensprünge, genannt Schrotrauschen.
- $\boldsymbol{c_{ij}}$ bezeichnet Rauschen, das statt durch Lichteinwirkung auf einen Pixel durch die Wärme der Photozelle entsteht, genannt Dunkelstrom.
- $\boldsymbol{\nu_{ij}}$ bezeichnet einen Term für weitere Rauschquellen, wie zum Beispiel Ungenauigkeiten beim Auslesen der Sensorzellen.

Alle Rauschquellen außer $\boldsymbol{f_{ij}}$ variieren von Bild zu Bild. Des weiteren gibt es bezüglich des PRNU-Rauschens die Besonderheit, dass es die ideale Intensität multiplikativ verändert. Damit ähnelt dieses Rauschmuster einem eingebetten Wasserzeichen (siehe Abschnitt 4.1.2, bzw. Gleichung 4.1), und wird in ähnlicher Weise für die Kameraidentifikation genutzt.

Dieses Rauschen kann unter Laborbedingungen direkt bestimmt werden und zwar mit einer Technik namens „Flat Fielding“: Mit Hilfe von Aufnahmen ohne Belichtungsquelle kann der Dunkelstrom $\boldsymbol{c_{ij}}$ ermittelt und herausgerechnet werden. Der Term $\boldsymbol{f_{ij}}$ kann nun ermittelt werden, indem die Szene perfekt uniform ausgeleuchtet wird. Über mehrere Aufnahmen hinweg variieren die verbleibenden Rauschterme, und können so durch Mittelwertbildung neutralisiert werden. Damit kann mit Hilfe von $\boldsymbol{k}$ Aufnahmen ein Schätzwert für die Photo Response Non-Uniformity $\hat{f}_{ij}$ pro Pixel berechnet werden als

$$\hat{f}_{ij} = \frac{\sum_k f_{ij}^{(k)}}{\frac{1}{m \cdot n} \sum_{i,j,k} f_{ij}^{(k)}} \quad . \tag{4.5}$$

In dieser Gleichung wird der Teiler über ein kleines Fenster von ***n*** mal ***m*** Pixeln gemittelt.

Normalerweise benutzt man Flat Fielding, um das Sensorrauschen aus einem Bild zu ***entfernen***. Dies wird aber auf Grund des hohen Aufwands nur für besondere Kameras vorgenommen, wie beispielsweise für Weltraumteleskope. Dazu wird die korrigierte Intensität $\hat{x}_{ij}$ berechnet als

$$\hat{x}_{ij} = \frac{y_{ij} - c_{ij}}{\hat{f}_{ij}} \quad . \tag{4.6}$$

Bei „normalen“ Kameras wird das PRNU-Rauschen von den Herstellern nicht kompensiert und kann zur forensischen Analyse herangezogen werden.

Für die forensische Ausnutzung des PRNU-Rauschens muss für die Bestimmung von $\hat{f}_{ij}$ also ein ähnlicher Ansatz gewählt werden wie für Flat Fielding — jedoch unter geschickter Vermeidung des aufwändigen Laboraufbaus. Dies ist im allgemeinen nur möglich, wenn Kompromisse bei der Präzision der PRNU-Schätzung eingegangen werden: Soll das PRNU-Muster eines bestimmten Bildes geschätzt werden, ist die abgebildete Szene im allgemeinen weder leer noch homogen ausgeleuchtet. Typischerweise wurden die Bilder auch auf unbekannte Art von dem digitalen Signalprozessor der Kamera nachverarbeitet, was beispielsweise bei Teleskopen ausgeschlossen werden kann.

Lukáš u. a. (2006) schlagen daher vor, den exakten Laboraufbau durch einen approximativen Ansatz aus der Signalverarbeitung zu ersetzen. Hierbei ist es nicht wichtig, für jeden Pixel den Faktor $\hat{f}_{ij}$ exakt zu bestimmen, sondern lediglich für eine ausreichend große Zahl der Pixel im Bild in etwa richtig zu schätzen. Hierin zeigt sich die Ähnlichkeit des Verfahrens zum Watermarking: es ist kein Problem, wenn einzelne Pixel „beschädigt“ sind, so lange das Muster als ganzes noch erkennbar ist.

Zur Simulation des Flat Fieldings wird die Annahme getroffen, dass die Frequenz des Rauschens im allgemeinen höher ist als die Frequenz des

Bildinhalts. Sei ***p*** ein Vektor, in dem alle Intensitäten des Bildes enthalten sind (also ein eindimensionales Datenfeld). Um aus ***p*** das Rauschmuster zu extrahieren, werden zuerst die tiefen Frequenzen entfernt: Sei ***F*(*p*)** ein Tiefpassfilter, der auf dem Bild ***p*** angewendet wird, dann ist der hochfrequente Anteil ***n*** des Bildes

$$n_i = p_i - F(p)_i \quad , \tag{4.7}$$

der ***i***-te Pixel in ***n*** ist also die Differenz des Bildes mit dem Tiefpass-gefilterten Bild. Farid (2011) empfiehlt für ***F*** einen Wavelet-basierten Entrauschungsfilter. Für erste Experimente kann jedoch auch einfach ein Wiener Filter benutzt werden.

Zur Entfernung der weiteren Rauschquellen muss Gleichung 4.7 auf eine größere Anzahl Bilder angewandt werden, die sicher aus der fraglichen Kamera stammen. Die hochpassgefilterten Rauschmuster enthalten dann durch Mittelung von 50 oder mehr Bildern einen relativ klaren digitalen „Fingerabdruck“ ***N*** der Kamera. ***N*** ist hierbei ein Vektor, der für jeden Eintrag von ***p*** einen Rauschwert enthält.

Um nun festzustellen, ob ein digitales Bild mit dieser Kamera aufgenommen wurde, wird das Bild mit unbekanntem PRNU ***P*** ebenfalls mit Gleichung 4.7 gefiltert. Das Rauschmuster wird direkt mit dem Fingerabdruck ***N*** korreliert: Sei $\bar{P}$ der Mittelwert aller Einträge von ***P*** , und $\bar{N}$ der Mittelwert aller Einträge von ***N*** , dann ist die Korrelation von ***N*** und ***P***

$$\mathrm{corr}(N,P) = \frac{\sum_{i=1}^{|N|}(N_i - \bar{N}) \cdot (P_i - \bar{P})}{\sqrt{\sum_{i=1}^{|N|}(N_i - \bar{N})^2} \cdot \sqrt{\sum_{i=1}^{|P|} P_i - \bar{P}}} \quad . \tag{4.8}$$

Die Korrelation dient also als Maßzahl, wie gut ***N*** und ***P*** zusammenpassen.

Wenn das Bild aus einer anderen Kamera stammt, sollte die Korrelation in etwa 0 sein (wir schreiben $|\mathrm{corr}(\boldsymbol{N}, \boldsymbol{P})| \leq \epsilon$). Für $\mathrm{corr}(\boldsymbol{N}, \boldsymbol{P}) > \epsilon$ wird angenommen, dass das Bild aus der selben Kamera stammt, aus welcher der Fingerabdruck ***N*** extrahiert wurde.

Analog zum Watermarking kann die grundlegende Methode noch weiter verfeinert werden: Beispielsweise kann das Rauschmuster nur über einer Maske

extrahiert werden, um die Zuverlässigkeit zu steigern. Experimentelle Ergebnisse zeigen, dass das Verfahren relativ resistent ist gegen externe Störungen, wie beispielsweise JPEG-Kompression und nachträgliche Aufhellung oder Abdunkelung des Bildes. Eine Schwäche des Verfahrens ist die Anfälligkeit gegen Desynchronisierungsangriffe: Wenn das Bild beispielsweise skaliert wurde, überdeckt der digitale Fingerabdruck nicht mehr eins-zu-eins die entsprechenden Sensorzellen. Ein weiteres schwieriges Problem ist das effiziente Finden des richtigen Fingerabdrucks aus einer sehr großen Kameradatenbank. Gäbe es beispielsweise eine „Registrierungspflicht“ für digitale Kameras, wäre nach derzeitigem Kenntnisstand der Aufwand, zu einem beliebigen Bild die dazu registrierte Kamera zu finden zu groß.

Exkurs 10 (Kamera-Fingerabdruck selbst berechnen) ***Es ist nicht schwierig, eigene Experimente mit dem Fingerabdruck einer Kamera zu unternehmen. Beispielsweise gibt Hany Farid in seinem Vorlesungsskript eine Implementierung dieses Verfahrens in Matlab an (Farid, 2011). Die Schritte für ein eigenes Experiment sind:***

1. ***Wählen Sie 50 bis 100 Bilder I_i, die aus der selben Kamera stammen, und mit den selben Zoom-Einstellungen aufgenommen wurden***.
2. ***Für jedes dieser Bilder, berechnen Sie die Differenz W_i aus Originalbild und geglättetem Originalbild***.
3. ***Der Fingerabdruck kann berechnet werden als***

$$f = \frac{\sum_i W_i \cdot I_i}{\sum_i I_i \cdot I_i} \ , \tag{4.9}$$

   wobei die Multiplikationen pixelweise durchgeführt werden.

4. ***Ein Bild J kann nun auf den Fingerabdruck getestet werden. Nach Berechnung der Differenzbilds W_J (analog zu W_i) weist die Korrelation auf den Fingerabdrucks hin. Für höhere Robustheit werden f und W_J jeweils noch mit J multipliziert:***

$$\mathrm{corr}(J \cdot f, J \cdot W_J) \ . \tag{4.10}$$

Beachten Sie, dass die Multiplikation elementweise durchgeführt werden muss, so dass die Ergebnisse wiederum Vektoren sind.

Testen Sie dies mit eigenen Bildern: Versuchen Sie, 20 ***eigene Bilder Ihrer Kamera zuzuordnen. Versuchen Sie, Ihre Bilder von*** 20 ***fremden Bildern zu unterscheiden***.

Abbildung 4.9: Illustration von JPEG-Kompressionsartefakten für `libJPEGs` Qualitätsstufen 100 und 20.

4.3 Inhaltliche Erkennung von Bildmanipulationen

Zwei unterschiedliche Ansätze erlauben Manipulationen direkt an Hand des Bildinhalts zu erkennen: Einerseits gibt es Algorithmen zur expliziten Suche nach Manipulationsspuren, beispielsweise nach kopierten, skalierten oder rotierten Regionen. Andererseits lassen sich im Bild physikalische und optische Gesetze auf Konsistenz prüfen, was besonders effektiv sein kann, wenn ein Bild aus mehreren anderen Bildern zusammengesetzt wurde. Wir betrachten hierfür einige Beispielmethoden, nämlich die Ausnutzung von JPEG-Artefakten, die Erkennung von kopierten und skalierten Regionen, und die Analyse der Beleuchtungsrichtung auf Objekten. Jedes der vorgestellten Verfahren geht von unterschiedlichen Grundannahmen aus. In der Praxis wird ein strittiges Bild mit allen zur Verfügung stehenden Werkzeugen untersucht. Von dem Standpunkt eines Fälschers betrachtet müssen also alle „Fehler“ vermieden werden, die unter den gegebenen Voraussetzungen zu einer Entdeckung führen würden. Durch die Entwicklung neuer und verbesserter forensischer Verfahren sollen derartige Schlupflöcher möglichst klein gemacht werden.

4.3.1 Spuren in Kompressionsartefakten

JPEG-Kompression ist das populärste Bildformat in heutigen Kompaktkameras und im Internet. Für die Kompression wird das Bild in Blöcke von 8 × 8 Pixeln unterteilt, die verlustbehaftet gespeichert werden. Die Stärke des Verlusts wird über den Qualitätsparameter bestimmt. Abbildung 4.9 illustriert dies für zwei Extremfälle: Während beispielsweise in der Implementierung der libJPEG bei Qualitätsstufe 100 der Qualitätsverlust fast unsichtbar ist, treten bei Qualitätsstufe 20 deutlich sichtbare Blockartefakte auf.

Die Speicherplatzeffizienz von JPEG-Bildern wird durch diese Informationsreduktion ermöglicht. Für viele bildforensische Verfahren stellt starke JPEG-Kompression ein Problem dar, wie zum Beispiel die Resampling-Erkennung (siehe Abschnitt 4.3.3). Dennoch läßt sich eine forensische Analyse auf JPEG-Bildern oftmals gut durchführen: Inkonsistenzen in den JPEG-Artefakten können ein manipuliertes Bild direkt enttarnen. Die Idee zur forensischen Ausnutzung von JPEG-Artefakten stammt von Lukáš u. Fridrich (2003). Wir betrachten hier jedoch nur die grundlegende Idee, und folgen dabei der etwas einfacheren Darstellung von Popescu u. Farid (2005b). Mittlerweile wurde in einer Vielzahl von Nachfolgearbeiten eine ganze Familie davon abgeleiteter, verbesserter Methoden aufgebaut.

Der JPEG-Kompressionsalgorithmus besteht im wesentlichen aus vier Schritten:

1. Das Bild wird in Blöcke von 8 × 8 Pixeln unterteilt
2. Jeder Block wird mit der diskreten Cosinus-Transformation (DCT) in den Frequenzraum übertragen.
3. Die Frequenzen werden durch unterschiedliche Quantisierungsfaktoren geteilt, höhere Frequenzen durch größere Faktoren als niedrige Frequenzen. Der Divisionsrest wird verworfen. Bei den hohen Frequenzen wird durch die Quantisierung also mehr Information abgeschnitten.
4. Die quantisierten Frequenzen werden zusammen mit den Quantisierungsparametern gespeichert.

Beispiel 26 (Kompressionsartefakte) ***Wir rechnen nun den Quantisierungsschritt nach. Bei der Kompression eines Bildes wird für alle Blöcke die gleiche Tabelle von Quantisierungsfaktoren benutzt. Nehmen wir***

an, die ersten vier Quantisierungsfaktoren seien (2, 3, 3, 4). ***Nehmen wir nun an, dass ein Block von*** 8 × 8 ***Pixeln DCT-transformiert wird. Die Ergebniskoeffizienten seien*** (4302, 2833, 2967, 2489). ***Jeder dieser Koeffizienten wird durch den entsprechenden Quantisierungsfaktor geteilt:***

$$4302/2 = 2151\ Rest\ 0 \tag{4.11}$$

$$2833/3 = 944\ Rest\ 1 \tag{4.12}$$

$$2967/3 = 989\ Rest\ 0 \tag{4.13}$$

$$2491/4 = 622\ Rest\ 3 \tag{4.14}$$

Die Divisionsreste werden ignoriert. Um das Bild zu speichern, wird erst die für das ganze Bild benutzte Quantisierungstabelle abgelegt, dann für jeden Bildblock das Divisionsergebnis, in unserem Beispiel die vier Koeffizienten (2151, 944, 989, 622).

Durch Vergrößerung der hinteren Quantisierungsfaktoren können hohe Frequenzen relativ einfach beschnitten werden. Je größer der Quantisierungsfaktor, desto kleiner (und ungenauer) der gespeicherte Koeffizient. Bei Benutzung typischer JPEG-Quantisierungstabellen ist das Divisionsergebnis der höchsten Frequenzen sehr oft 0.

Für die Dekompression werden die Schritte rückwärts abgelaufen, allerdings ist der Informationsverlust durch die Quantisierung in Schritt 3 unumkehrbar — die Qualität des Bildes nimmt durch das Speichern ab. Für jeden Block wird die selbe Quantisierungstabelle benutzt. Mit welchen Werten quantisiert wird hängt einerseits von der Implementierung des Kompressors ab (jeder Hersteller wählt seine eigene Quantisierungstabelle), andererseits von der Qualitätsstufe der Kompression. Sei ***u*** ein DCT-Koefftizient in einem Block (stellvertretend für eine Frequenz), und ***a*** der zugehörige Quantisierungsfaktor. Dann wird der Wert

$$q_a(u) = \left\lfloor \frac{u}{a} \right\rfloor \tag{4.15}$$

gespeichert, wobei $\lfloor \cdot \rfloor$ der Abrundungsoperator ist. Wird nun ein JPEG-Bild geöffnet, und ein Block ein zweites mal mit einem anderen Koeffizienten ***b*** gespeichert, ergibt sich durch doppelte Rundung der Koeffizient

$$q_{ab}(u) = \left\lfloor \left\lfloor \frac{u}{a} \right\rfloor \frac{a}{b} \right\rfloor \quad . \tag{4.16}$$

Wir illustrieren den Unterschied zwischen einfacher und doppelter Quantisierung mit einem Zahlenbeispiel. Sei der Quantisierungsfaktor für die erste Kompression 2, für die zweite Kompression 3. Des weiteren nehmen wir an, dass für die Koeffizienten ***u*** die Werte 0 bis 13 auftreten. Das Zahlenbeispiel in Tabelle 4.1 zeigt, dass bei einfacher JPEG-Kompression (siehe Gleichung 4.15) die Koeffizienten gleichverteilt auftreten, während bei Doppelquantisierung (siehe Gleichung 4.16) die Verteilung geändert wird: Nun treten 0 und 2 doppelt so häufig auf wie 1 und 3. Dies funktioniert allerdings nur, wenn der zweite Quantisierungsfaktor kein Vielfaches des ersten ist.

Koeffizient ***u***	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Quant, ***a*** = 2	0	0	1	1	2	2	3	3	4	4	5	5	6	6
Dopp.-Quant. 2, 3	0	0	0	0	1	1	2	2	2	2	3	3	4	4

Tabelle 4.1: Doppelquantisierung mit unterschiedlichen Quantisierungsfaktoren ändert die Verteilung der Koeffizienten.

Dieser Effekt kann direkt forensisch ausgenutzt werden: Durch zählen der Koeffizienten in benachbarten Bildblöcken kann ein Histogramm wie in Tabelle 4.1 aufgestellt werden. Die stark unterschiedliche Verteilung im Fall von Doppelquantisierung kann über eine Frequenztransformation des Histogramms als unübliche Häufung („Spitze“) im hohen Frequenzbereich vollautomatisch erkannt werden.

Der Umstand, dass ein Bild zweimal im JPEG-Format gespeichert wurde, ist für sich kein Beweis für eine Fälschung. Interessanter wird es allerdings, wenn lediglich ein ***Teil*** des Bildes doppelt komprimiert ist, während ein anderer Teil nur einfach komprimiert ist. Dies geschieht zum Beispiel, wenn in einem JPEG-Bild mit einem Zeichenpinsel gearbeitet wird: Durch die Zeichenoperation werden die JPEG-Artefakte der Zeichenregion zerstört. Bei nochmaligem Speichern im JPEG-Format ist der unveränderte Hintergrund doppelt komprimiert, während die retuschierte Region lediglich einfach komprimiert erscheint. Damit steht ein starkes Werkzeug zur Verfügung, um JPEG-Bilder zu analysieren.

4.3.2 Erkennung von Copy-Paste-Fälschungen

Wenn Bildinhalt verdeckt werden soll oder eine unstrukturierte Menge in einem Bild vermehrt werden soll, greifen Bildbearbeiter oftmals zu dem Klon-Pinsel. Hierbei wird eine Komponente innerhalb des Bildes teilweise oder komplett kopiert. Ein Beispiel ist in der Werbeanzeige in Abbildung 4.10 gezeigt (erschienen im Spiegel (2010)). Die unstrukturierte Masse der Fußballfans (links) ist ein typisches Ziel für Copy-Move-Manipulationen. In der Mitte sind die kopierten Regionen farbig eingezeichnet (Teile der Schrift wurden irrtümlich auch markiert). Im rechten Bild heben die hellen Regionen den kopierten Teil besonders hervor.

Die Erkennung derartiger sogenannter „Copy-Paste-“ bzw. „Copy-Move-Fälschungen“ ist das am intensivsten untersuchte Teilgebiet der Bildforensik. Da der kopierte Inhalt zweimal in dem Bild enthalten ist, muss im wesentlichen nur die Selbstähnlichkeit zwischen diesen Regionen gefunden werden.

Die generelle Herangehensweise ist, aus der Nachbarschaft um jeden Pixel im Bild einen Merkmalsvektor zu berechnen. Aus der Menge aller Merkmalsvektoren werden Paare besonders ähnlicher Vektoren näher betrachtet. Weisen mehrere Paare ein gemeinsames Verhalten auf — beispielsweise die selbe Entfernung oder den selben Versatz zu ihrem Partner — werden diese als kopierte Regionen markiert.

Die Herausforderungen zur effektiven Erkennung von Copy-Paste-Fälschungen liegen im wesentlichen in zwei Punkten: Die Wahl eines Merkmalsvektors, der sowohl resistent gegen zusätzliche Störungen wie Rauschen oder JPEG-Kompression ist, und die Reduktion der Rechenzeit für die Berechnung der Merkmale und ihrer Gruppierung. Beides sind entgegengesetzte Ziele: aussagekräftigere Merkmale sind im allgemeinen rechenaufwendiger.

Beispielsweise wurde von Ryu u. a. (2010) vorgeschlagen, so genannte Zernike-Momente als Merkmale zu berechnen. Dabei handelt es sich um statistische Momente, die zu einem bestimmten Grad auch rotations- und skalierungsinvariant sind. Dies ist vorteilhaft, um nicht nur direkte Kopien zu erkennen, sondern auch solche, in denen die kopierte Region in das Bild eingepasst werden musste. Die Kopien in Abbildung 4.10 wurden ebenfalls mit Zernike-Momenten annotiert. Bei allen positiven Eigenschaften sind die Anforderungen an die Rechenzeit erheblich. Die Analyse auf einem einzelnen Bild von 2000 × 3000 Pixeln kostet auf einem aktuellen Arbeitsplatzrechner über eine Stunde Rechenzeit.


Sie lassen auch
2010 ganz
Deutschland mitfiebern.
Weil Sie Gebühren zahlen.
ARD
ZDF
UND SIE.


Sie lassen auch
2010 ganz
Deutschland mitfiebern.
Weil Sie Gebühren zahlen.
ARD
ZDF
UND SIE.

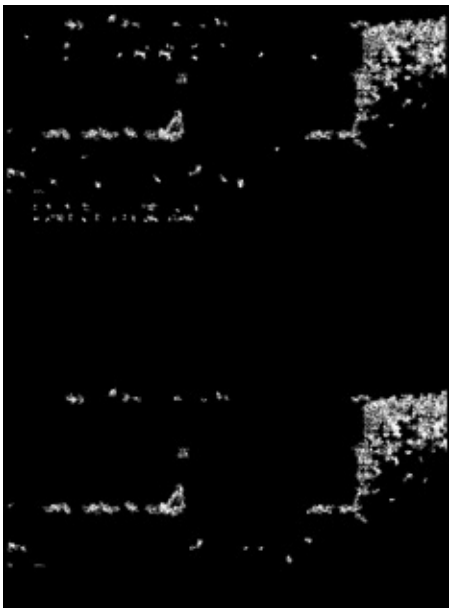


Abbildung 4.10: Werbeanzeige mit kopiertem Bildinhalt (aus Spiegel (2010)).

Um den Rechenaufwand zu reduzieren, schlugen verschiedene Autoren (beispielsweise Pan u. Lyu (2010); Amerini u.a. (2011)) die Nutzung von SIFT-Merkmalen vor. SIFT-Merkmale („Scale-Invariant Feature Transform“, siehe Lowe (2004)) können ebenfalls mit skalierten und rotierten Kopien umgehen. Die eigentliche Reduktion der Rechenzeit kommt jedoch von dem Umstand, dass SIFT-Merkmale nicht um jeden Bildpunkt herum berechnet werden, sondern lediglich auf „informativen“ Bildregionen, beispielsweise Ecken oder Kanten. Dies wird mit zwei Nachteilen erkauft: einerseits werden Kopien homogener Regionen typischerweise übersehen, andererseits wird eine flächige Markierung der Kopie erschwert, da die Berechnung nur auf vereinzelten markanten Punkten durchgeführt wird. In der Praxis wird dies aber sehr gut kompensiert durch die deutlich kürzere Rechenzeit: Die Ergebnisse für ein Bild liegen innerhalb weniger Minuten vor.

4.3.3 Erkennung von Skalierung und Rotation

Bei der Erstellung von Bildcollagen ist es häufig nötig, die Größe und Ausrichtung des eingefügten Objekts anzupassen. In Bildverarbeitungsprogrammen sind diese Anpassungen als Interpolationsmethoden implementiert. Wird etwa ein Bild auf seine doppelte Größe skaliert, muss jeder zweite Pixel aufgefüllt werden. Je nach

Interpolationsverfahren kann dies beispielsweise eine gewichtete Summe der Nachbarn sein („bilineare Interpolation“), oder einfach die Farbe eines Nachbarpixels („Nearest-Neighbor Interpolation“). Damit wird ein Zusammenhang hergestellt zwischen den aufgefüllten Pixeln und ihren Nachbarn im Bild.

Spezialisierte Methoden zur Erkennung von Skalierung und Rotation nutzen genau diesen Zusammenhang aus. Das erste Verfahren zur Erkennung von Interpolationsspuren wurde von Popescu und Farid vorgestellt (siehe Popescu u. Farid (2005a)), weitere wegweisende Arbeiten stammen von Gallagher, Chen und Kirchner (siehe Gallagher u. Chen (2008), Kirchner (2010)).

Wir betrachten den Ansatz von Popescu und Farid genauer. In einer nicht-interpolierten Region wird angenommen, dass benachbarte Pixel voneinander weitestgehend unabhängig sind. Das heißt, dass kein systematisches Muster zwischen benachbarten Bildpunkten existiert. In einer interpolierten Bildregion wird hingegen ein linearer Zusammenhang zwischen benachbarten Bildpunkten angenommen. Dieser Zusammenhang ist nicht beliebig, stattdessen tritt die selbe Beziehung periodisch wiederkehrend auf: In dem einführenden Beispiel von einer Skalierung auf die doppelte Größe wird alle zwei Pixel ein Bildpunkt aus seinen Nachbarn gewichtet. Wird eine Region beispielsweise auf 75% der Originalgröße verkleinert, so werden aus 4 Originalpixeln je 3 Zielpixel. In einem periodischen Abstand können einzelne Pixel also als gewichtete Summe ihrer Nachbarpixel dargestellt werden. Aber mit welchen Gewichten wird diese Summe gebildet? Diese Information steht dem Forensiker bei der Analyse nicht zur Verfügung. Popescu und Farid schlagen vor, die Gewichte mit dem „Expectation-Maximization“-Algorithmus zu schätzen. Dies ist ein iteratives Verfahren aus der Mustererkennung. Als Initialisierung wird eine Anfangsgewichtung zufällig geraten. Diese Anfangsgewichte werden in den folgenden Schritten korrigiert, bis eine hinreichend gute Lösung gefunden ist. Hierbei wird für jeden Pixel eine Wahrscheinlichkeit geschätzt („***p***-Map“), dass er mit diesen Gewichten von seinen Nachbarn abhängig ist. Im zweiten Schritt wird die Gewichtung optimiert. Hierbei haben die Pixel mehr Einfluss, denen im ersten Schritt eine höhere Wahrscheinlichkeit zugeordnet wurde. Diese Schritte werden wiederholt, bis die Gewichte sich nicht mehr wesentlich verändern. Die zugeordneten Wahrscheinlichkeiten, p-Map genannt, werden für die weitere Analyse benutzt. Wenn ein (periodisches) Interpolationsmuster vorliegt, erscheint es im Frequenzraum als isolierte Spitze. Wenn das Zielbild zusätzlich im JPEG-Format gespeichert wurde, wird die Erkennung der Interpolation

schwieriger: Die ebenfalls periodischen JPEG-Artefakte überlagern häufig die Interpolationsspuren im Bild.

Abbildung 4.11 zeigt ein Beispiel für das Verfahren. Oben links ist das Originalbild gezeigt, unten links das selbe Bild mit einer auf 140% skalierten eingefügten Blüte. Wir nehmen an, dass das Bild links unten eine Fälschung sei. Wir verfolgen nun zwei Fälle. Im ersten Fall wurde das Bild in einem verlustfreien Format gespeichert (nehmen wir an als PNG-Bild), im zweiten Fall im verlustbehafteten JPEG-Format, mit Qualitätsstufe 90. Die mittlere Spalte zeigt die p-Maps für beide Varianten, für das PNG-Bilds in der oberen Reihe und das JPEG-Bild in der unteren Reihe. Es zeigt sich, dass der helle Bereich, der die Skalierung anzeigt, in dem unteren Bild wesentlich kontrastärmer ausgeprägt ist. Dies kommt daher, dass JPEG-Kompression die Erkennung der Resampling-Artefakte erschwert. Diese Eigenschaft lässt sich zusätzlich in der rechten Spalte beobachten. Hier wurden die beiden p-Maps in den Frequenzraum transformiert, da dort die periodischen Interpolationsspuren als kleine Spitzen erkennbar sind. Der interessante Bereich ist orange eingekreist. Die Position der Spitzen variiert mit dem Grad der Skalierung oder der Rotation. In dem PNG-Bild tritt die Frequenzspitze aus der Interpolation deutlich hervor. In dem JPEG-Bild ist sie jedoch nicht mehr zu erkennen.

Abbildung 4.11: Beispiel für die Erkennung von Skalierung nach Popescu und Farid (Popescu u. Farid, 2005a). Mit JPEG-Kompression wird die Erkennung der Skalierung schwieriger, sowohl in der ***p***-Map als auch in dem Fourier-Spektrum.

Abgesehen von der Anfälligkeit des Verfahrens gegen JPEG-Kompression ist Interpolations-Erkennung im allgemeinen fast immer anwendbar. Die Ergebnisse sind oft aussagekräftig und gut interpretierbar. Neuere Verfahren nutzen

ähnliche Ideen, sind jedoch etwas robuster und recheneffizienter in der Anwendung.

4.3.4 Szenenkonsistenz

Die bisher betrachteten Verfahren basierten auf statistischen Argumenten, entweder durch Ausnutzung des Bildformats oder spezifischer Spuren aus der Erstellung der Fälschung. Im Kontrast hierzu betrachten wir nun ein physikbasiertes Verfahren zur Überprüfung der Lichtrichtung. Das menschliche Auge ist relativ unsensibel bezüglich leicht unterschiedlichem Lichteinfall. Daher ist es nicht notwendig (und abgesehen davon auch nicht einfach) eine Fälschung mit absolut korrekter Beleuchtung zu erstellen, denn ein menschlicher Betrachter sieht typischerweise auch halbwegs korrekten Lichteinfall als richtig an. Johnson u. Farid (2007) haben jedoch ein Verfahren vorgestellt, mit dem die Richtung des Lichteinfalls in der Bildebene explizit berechnet und geprüft werden kann. Die Idee zur forensischen Ausnutzung ist schnell skizziert (die Berechnung ist allerdings relativ kompliziert): In einer Szene, in der die Lichtquelle sehr weit entfernt ist (beispielsweise die Sonne), fallen die Lichtstrahlen ungefähr parallel auf alle Objekte in der Szene. Um die Konsistenz der Szene zu bewerten, werden die Lichteinfallsrichtungen auf je zwei Objekten in der Szene miteinander verglichen.

Die Methode benutzt einige Konzepte der physikbasierten Bildverarbeitung. Für die Berechnung der Lichtrichtung werden auf den zu untersuchenden Objekten die Oberflächennormalen geschätzt. Die Oberflächennormale ist der Richtungsvektor, der senkrecht auf der Oberfläche steht (abhängig von dem momentan betrachteten Punkt). Wie in der rechten Hälfte von Abbildung 4.12 gezeigt, ist der dreidimensionale Normalenvektor in einem Bild im allgemeinen unbekannt. Die Kernidee von Johnson und Farid beruht auf dem Umstand, dass entlang einer Objektkontur im Bild die Oberflächennormale in der Bildebene liegt. Beispielsweise entspricht die 3D-Normale in etwa der Senkrechten der Objektkontur entlang der Schulter in Abbildung 4.12 (links).

Abbildung 4.12: Normalen entlang Objektkonturen, z.B. entlang der Schulter der linken Person, liegen in der Bildebene. Sie können aus dem Konturverlauf berechnet werden. Normalen an beliebigen Punkten, z.B. auf dem Handgelenk der rechten Person, ragen in den Raum. Sie sind aus einem 2D-Bild in der Regel nicht zuverlässig schätzbar.

In dem theoretischen Modell wird die Lichtmenge, die von einen Punkt $E(\vec{N})$ mit Normale $\vec{N}$ reflektiert wird als

$$E(\vec{N}) = \int_{\Omega} L(\vec{V}) R(\vec{V}, \vec{N}) d\Omega \tag{4.17}$$

modelliert. Hierbei bezeichnet $L(\vec{V})$ die Menge des einfallenden Lichts aus Richtung $\vec{V}$ und $R(\vec{V}, \vec{N})$ die Reflexionsfunktion der Oberfläche. Das Integral wird über alle Raumwinkel Ω berechnet.

Johnson und Farid zeigen, dass unter der Annahme von diffuser Reflexion, homogener Objektfarbe und unter Ausschluss von Schatten ein lineares Gleichungssystem aufgestellt werden kann, mit dem in der Bildebene der Helligkeitsverlauf des Lichteinfalls aus verschiedenen Richtungen geschätzt wird. Ein Benutzer muss die (gleichfarbigen, nicht schattigen) Konturen des Objekts markieren. Der Helligkeitsverlauf an den Konturen und die geschätzten Oberflächennormalen dienen als Eingabe für das Gleichungssystem. Die

Verteilung des Lichteinfalls wird in Kugelkoordinaten, „Spherical Harmonics“, dargestellt. Dann wird die Beleuchtungsumgebung approximiert als

$$L(\vec{V}) = \sum_{n=0}^{2} \sum_{m=-n}^{n} l_{n,m} Y_{n,m}(\vec{V}) \quad . \tag{4.18}$$

Hierbei bezeichnet $\boldsymbol{Y_{n,m}}$ die $\boldsymbol{n}$, $\boldsymbol{m}$-te Basisfunktion der Spherical Harmonics, und $\boldsymbol{l_{n,m}}$ den gesuchten Beleuchtungskoeffizienten.

Nachdem $\boldsymbol{l_{n,m}}$ für jedes Objekt ermittelt wurde, kann die Korrelation zwischen diesen Koeffizienten berechnet werden. Hierbei steht eine niedrige Korrelation für unterschiedliche Beleuchtungsumgebungen.

Wie ist dieser Ansatz zu bewerten? In einigen Fälschungsszenarien bringt dieses forensischen Verfahren den Fälscher in große Schwierigkeiten: Angenommen, er möchte eine Person des öffentlichen Lebens in einer kompromittierenden Situation abbilden, dann stehen ihm möglicherweise nur wenige Quellbilder zur Verfügung, um die Szene plausibel zusammenzufügen. Sollte die Beleuchtungssituation des eingefügten Objekts mit der Lichtsituation der Szene nicht zufällig übereinstimmen, kann der Fälscher dies lediglich durch aufwändiges Retuschieren beheben; dies birgt jedoch die Gefahr, dass er andere Fehler in das Bild einfügt. Diese Vorteile kommen allerdings zu einem stattlichen Preis: Die Implementierung des Verfahrens ist vergleichsweise kompliziert, und bis dato sind die Annahmen, die das Verfahren an die Szene stellt, relativ restriktiv. Insbesondere der Ausschluss von Schatten, und die Anforderung an Konturen entlang des selben Objektmaterials sind in der Praxis oft schwierig zu erfüllen. Diese Anforderungen aufzubrechen ist eines der aktuellen Forschungsprobleme.

4.4 Werkzeuge für Forschung und Entwicklung

Einige Forschungsgruppen stellen für ihre entwickelten Methoden prototypische Software bereit, um die Methoden nachzuvollziehen, oder um den Vergleich mit anderen Arbeiten zu erleichtern. Zum Beispiel haben Hayati u. a. (2007) eine Übersicht von verfügbarer Steganographie-Software zusammengestellt. Oftmals ist es jedoch nötig, selbst Hand anzulegen, und bestehende Lösungen auf die eigenen Bedürfnisse anzupassen, oder neue Algorithmen selbst zu

implementieren. Wir betrachten kurz zwei populäre Bildverarbeitungsplattformen, die für diese Aufgabe genutzt werden können. Matlab ist ein proprietäres Programm, häufig genutzt für „Prototyping“, d.h. für die schnelle Entwicklung von Proof-of-Concept Implementierungen. OpenCV ist eine freie C/C++-Bibliothek für Bildverarbeitung und Computer Vision-Algorithmen. Die systemnahe, recheneffiziente Implementierung vieler Methoden erlaubt die plattformübergreifende Entwicklung größerer Softwaresysteme. Die Wahl des Werkzeugs hängt somit von dem Einsatzzweck der Software ab (Prototyp oder Produktivsystem), und den finanziellen Randbedingungen (Softwareprodukte vs. freie Software).

4.4.1 Prototyping

Vor allem in der Forschung ist es wichtig, innerhalb kurzer Zeit mathematisch komplexe Lösungen in Form eines Software-Prototypen realisieren zu können. Hierbei steht nicht der saubere Entwurf der Software, beispielsweise hinsichtlich Wiederverwendbarkeit oder Wartbarkeit im Vordergrund. Stattdessen ist das Ziel, schnell verschiedene Funktionalitäten zu akkumulieren, und auf den Eingabedaten zu testen („Prototyping“).

Ein sehr populäres Prototyping-Werkzeug ist Matlab (siehe Mathworks-Deutschland (2012)). Matlab beinhaltet einen Interpreter für die gleichnamige Programmiersprache und eine Entwicklungsumgebung mit Debugger. Zusätzliche Funktionalität kann in sogenannten „Toolboxes“, d.h. Zusatzbibliotheken, erworben werden. Die Originalsoftware von Mathworks ist relativ kostspielig, insbesondere für Unternehmen. Es gibt allerdings eine freie Implementierung mit ähnlicher Funktionalität mit dem Namen octave (siehe Octave (2012)).

Der Schwerpunkt von Matlab liegt auf dem Lösen numerischer Probleme, insbesondere linearer Gleichungssysteme. Entsprechend ist das vielleicht wichtigste Primitiv der Programmiersprache die Matrix. Dies geht so weit, dass Operationen, die auf Matrizen definiert sind, im allgemeinen effizienter ausgeführt werden als dieselbe Operation iteriert auf allen Elementen der Matrix.

Die Nutzung von Matlab bietet neben seinen Eigenschaften als Prototyping-Werkzeug noch einen zweiten Vorteil. Wenn Forscher Quellcode veröffentlichen, ist dieser häufig in Matlab geschrieben (siehe zum Beispiel die Codebeispiele in Farids Skript, die öffentlich verfügbar sind (Farid, 2011)), und kann häufig auch direkt in Matlab verwendet werden.

4.4.2 Applikationsentwicklung

Die Entwicklung rechenintensiver Algorithmen oder eines größeren Software-Frameworks ist in Matlab eher ungünstig. Oftmals empfiehlt es sich, auf eine systemnähere Plattform zu wechseln. Im Bereich der Bildverarbeitung ist die OpenCV-Bibliothek (siehe OpenCV (2012)) eine gute Wahl für ein C- bzw. C++-basiertes Software-Framework.

OpenCV wird unter einer freien Lizenz vertrieben, kann also kostenfrei genutzt werden. Die C-Schnittstelle ist relativ alt, die C++-Schnittstelle wird jedoch seit einigen Jahren intensiv weiterentwickelt. In der reichhaltigen Funktionssammlung sind unter anderem Klassifikatoren, Filter für die Bildverarbeitung, Standardoperationen auf Signalen wie Singulärwertzerlegung und die diskrete Fouriertransformation. Das Einlesen, Anzeigen und Speichern von Bildern ist ebenfalls in der Bibliothek gekapselt, so dass keine weiteren Low-Level Bibliotheken (wie zum Beispiel `libjpeg`) direkt angesprochen werden müssen. Die Bibliothek kann unter Windows und Linux genutzt werden.

Die Nachteile gegenüber Matlab sind die etwas längere Einarbeitungszeit und der Umstand, dass C bzw. C++ kompilierte Sprachen sind, also im allgemeinen schlechter geeignet sind für die Prototypenentwicklung.

---

[5] Siehe z.B. Sachnev u. a. (2009), oder Cheddad u. a. (2010) für eine allgemeine Einführung.

Kapitel 5

Vorgehensmodelle

Autoren: Andreas Dewald, Felix Freiling

Bei der Sicherung und Analyse digitaler Spuren muss eine allgemein akzeptierte und erprobte Vorgehensweise angewandt werden, damit diese Spuren in möglichst überzeugender Form als Beweismittel vor Gericht verwendet werden können. In diesem Kapitel befassen wir uns mit derartigen etablierten Vorgehensweisen, den sogenannten ***Vorgehensmodellen***. Ihr Name impliziert eine gewisse Abstraktheit: Vorgehensmodelle bilden also lediglich einen Handlungsrahmen, an dem sich das eigene Vorgehen orientieren kann.

Wie dieses Kapitel zeigen wird, gibt es in der Literatur viele Vorgehensmodelle. Die meisten dieser Modelle beschäftigen sich mit dem Vorgehen im Anschluss an einen konkreten Sicherheitsvorfall, etwa einen erkannten Angriff auf ein Firmennetzwerk. Wir werden zwei charakteristische Vertreter vorstellen: das ***Incident-Response-Modell*** von Mandia u.a. (2003) sowie den ***investigativen Prozess*** von Casey (2004). Beide eignen sich gut, um die wesentlichen Aspekte derartiger Vorgehensmodelle zu beschreiben, die anschließend im ***common model*** von Freiling u. Schwittay (2007) integrierend betrachtet werden. Ein Überblick über weitere Vorgehensmodelle in der Literatur schließt dieses Kapitel ab.

Die Beschreibungen und Diskussionen in diesem Kapitel sollen zeigen, dass es kein „bestes“ Vorgehensmodell gibt, sondern dass es vielmehr darauf ankommt, überhaupt ein Vorgehensmodell zu haben. Auch ist das Anwendungsfeld der forensischen Informatik breiter als die bloße Behandlung von Sicherheitsvorfällen — man denke etwa an solch gegensätzlichen Fälle wie den Nachweis des Besitzes von Kinderpornographie auf der einen Seite und einen Fall von Betrug oder Untreue in einem internationalen Konzern auf der anderen. Vorgehensmodelle eignen sich auch nicht, um konkrete technische Auswertungsschritte festzuhalten. In der Praxis werden derartige

Vorgehensmodelle deshalb regelmäßig ergänzt durch Prozessbeschreibungen und Checklisten, die versuchen, das Wissen über den Verfahrensablauf zu formalisieren und zu dokumentieren.

Im Detail verläuft jede Untersuchung jedoch anders, so dass zu genaue Vorgehensmodelle auch den Blick auf die Möglichkeiten einengen können. Ein gutes Vorgehen basiert also neben einem passenden Vorgehensmodell immer auch auf Erfahrung und Intuition.

5.1 Hypothesenbasiertes Arbeiten

Bereits in Kapitel 1 wurde eine grundlegende Einsicht wissenschaftlichen Arbeitens dargelegt, nämlich dass jede Beobachtung oder Analyse Fehler enthalten kann. Daraus entspringt die Notwendigkeit, sich und das eigene Vorgehen ständig zu hinterfragen. Dies geschieht durch die Formulierung falsifizierbarer Hypothesen. Diese Hypothesen bilden auch die Grundlage für das Vorgehen bei forensischen Analysen.

Hierzu werden zunächst viele Hypothesen als Teil einer Theorie des Tathergangs entwickelt. Anschließend wird versucht, einzelne Hypothesen zu widerlegen. Aus den verbleibenden Hypothesen können dann neue Hypothesen gebildet werden. Dies ist in Abbildung 5.1 dargestellt. Aus den nicht ausgeschlossenen initialen Hypothesen werden über die Zeit neue Hypothesen gebildet und stetig verfeinert, bis am Ende der Untersuchung eine Reihe von Hypothesen übrig bleiben, die auch mit größerem Aufwand nicht zu widerlegen sind. Diese Hypothesen werden dann als wahrscheinlich angenommen. Der Vorteil dieser Vorgehensweise ist, dass es schwerer ist, von einer konkreten Hypothese eingenommen zu werden. Daher ist eine höhere Wahrscheinlichkeit gegeben, dass man so objektive Ergebnisse erhält. Intuitiv entspricht die Anzahl der in jedem Schritt parallel aufgestellten Hypothesen der ***Breite*** einer Untersuchung, die Anzahl der aufeinanderfolgenden (verfeinerten) Hypothesen entspricht ihrer ***Tiefe*** (siehe Abbildung 5.1).

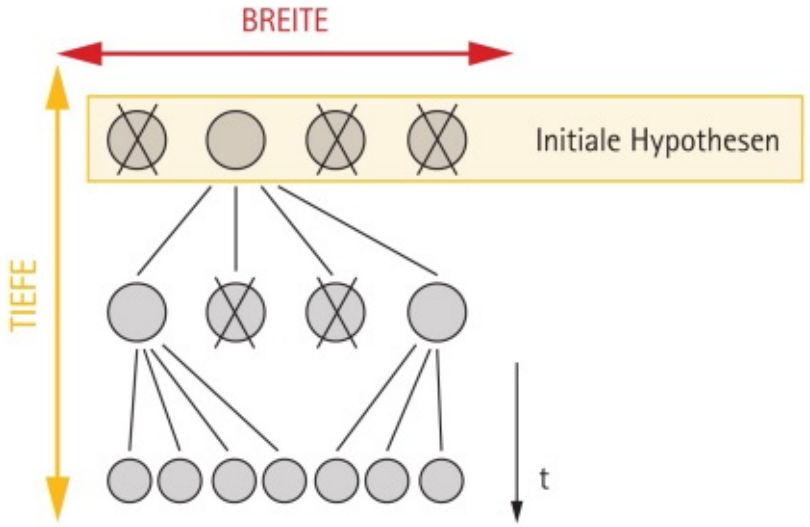


Abbildung 5.1: Breite vs. Tiefe einer hypothesenbasierten, forensischen Untersuchung.

Beispiel 27 (Hypothesenbasiertes Vorgehen) ***Als Beispiel betrachten wir einen Fall, in dem eine Festplatte beschlagnahmt wurde. Der Verdacht lautet auf Besitz und/oder das Verbreiten kinderpornographischer Dateien. Die initiale Hypothese lautet:***

Hypothese 1 ***Es sind keinerlei kinderpornographische Dateien auf dem Rechner vorhanden***.

Falsifizierung ***Der Ermittler findet entsprechende Dateien über einen auf Hashwerten basierenden Vergleich. Dies führt zu weiteren Hypothesen***.

Hypothese 2 ***Über den Rechner wurden zu keiner Zeit kinderpornographische Dateien verbreitet***.

Falsifizierung ***Der Ermittler findet die entsprechenden Dateien in einem freigegebenen Austauschordner eines Cloud-Speicher-Anbieters. Es kann also nicht ausgeschlossen werden, dass die Dateien über den Rechner verbreitet wurden***.

Die Effektivität einer Untersuchung hängt entscheidend von der ***Objektivität*** der Ermittler ab. Diese beginnen sofort, Theorien über den Tathergang zu bilden. Jeder Fall ist jedoch neu und einzigartig. Wichtig für eine objektive Beurteilung sind Fakten, nicht Vermutungen. Aus der kriminalistischen Praxis ist diese Gefahr als sogenannte ***Erfahrungsfalle*** bekannt: Wenn ein neuer Fall ähnlich zu

einem alten erscheint, ist man geneigt, den neuen mit denjenigen Mitteln anzugehen, die beim alten Fall zum Erfolg führten (Casey, 2004, S. 93).

Voreilige Theorien können dazu führen, dass bestimmte Spuren nicht mit der nötigen Sorgfalt untersucht oder falsch interpretiert werden, wie das folgende Beispiel von Casey (2004) zeigt.

Beispiel 28 (Risiken der Voreingenommenheit) ***Es wird eine gelöschte Datei gefunden, die ein nacktes Kleinkind zeigt. Bei der Datenwiederherstellung konnte das erste Zeichen des Dateinamens nicht wiederhergestellt werden (wie es typischerweise bei der Rekonstruktion gelöschter Dateien eines FAT-Dateisystems der Fall ist). Auf dem Bildschirm wird die Datei als*** ***`?orn1yr5.gif`*** ***angezeigt***.

Im Laufe der Ermittlung könnte man geneigt sein, den Originalnamen der Datei als ***`porn1yr5.gif`*** ***statt*** ***`born1yr5.gif`*** ***zu wählen. Beide legen unterschiedliche Vermutungen nahe. In einem solchen Fall ist es offensichtlich besser, ein neutrales Zeichen zu verwenden (z. B.*** ***`_orn1yr5.gif`*** ***) und zu dokumentieren, dass das erste Zeichen zerstört war***.

Das hypothesenbasierte Vorgehen ist universell geeignet, größtmögliche Objektivität bei einer forensischen Untersuchung zu gewährleisten. Im Folgenden werden konkrete Vorgehensmodelle beschrieben, die einen Rahmen für derartige Untersuchungen bilden.

5.2 Das Incident-Response-Modell

Das erste Vorgehensmodell, das wir näher betrachten werden, ist das ***Incident-Response-Modell***, das einen starken Fokus auf das Management eines Sicherheitsvorfalls legt. Zunächst müssen wir jedoch definieren, was genau ein Sicherheitsvorfall ist und was ***Incident Response*** eigentlich bedeutet.

5.2.1 Incident Response

Das NIST definiert einen ***computer security incident*** als „violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices“ (Grance u.a., 2004). Die deutsche Übersetzung von ***computer security incident*** ist üblicherweise ***IT-Sicherheitsvorfall***.

Organisationen müssen auf derartige Vorfälle in wohldefinierter Weise reagieren, da unvorbereitete und chaotische Reaktionen oft mehr Schaden anrichten als der eigentliche Vorfall. Aus diesem Grund werden Vorfälle üblicherweise von speziell hierfür gebildeten Teams, sogenannten ***computer security incident response teams*** (CSIRT), bearbeitet. ***Incident response*** ist die Reaktion einer Organisation auf einen Sicherheitsvorfall.

Obgleich sich das Wort ***incident*** als Sicherheitsvorfall ins Deutsche übersetzen lässt, werden wir der Einfachheit halber weiterhin den englischen Begriff ***incident response*** verwenden, da er in der einschlägigen Literatur ein feststehender Begriff und die deutsche Übersetzung „Reaktion auf einen Sicherheitsvorfall“ ein wenig sperrig ist.

5.2.2 Das Vorgehensmodell

Um in der Lage zu sein, auch komplexe Vorfälle korrekt zu behandeln, wird der Prozess des Incident Response in eine Reihe logischer Schritte unterteilt. Im Folgenden betrachten wir daher das entsprechende Modell nach Mandia u.a. (2003).

Incident Response muss systematisch und gut organisiert erfolgen. Bei großen Organisationen gibt es oft ein festes CSIR-Team, während bei kleinen Organisationen solche Teams häufig bei Bedarf zusammengestellt werden. CSIRTs erlauben im Regelfall eine schnelle Bestätigung, ob ein Vorfall tatsächlich geschehen ist, die schnelle Erkennung und Eindämmung des Vorfalls, professionelle Sicherung von Spuren, professionelle Dokumentation sowie die Verbesserung des Schutzes vor zukünftigen Vorfällen.

CSIRTs stehen oft unter erheblichem Druck: Sicherheitsvorfälle können die Existenz der Organisation gefährden. Ein Vorgehensmodell für Incident Response gibt dem CSIRT einen Leitfaden an die Hand, um möglichst wenige Fehler zu begehen. Das Vorgehensmodell von Mandia u. a. (2003) besteht aus insgesamt sieben Phasen, die in Abbildung 5.2 dargestellt sind. Wir gehen im Folgenden auf die einzelnen Phasen näher ein.

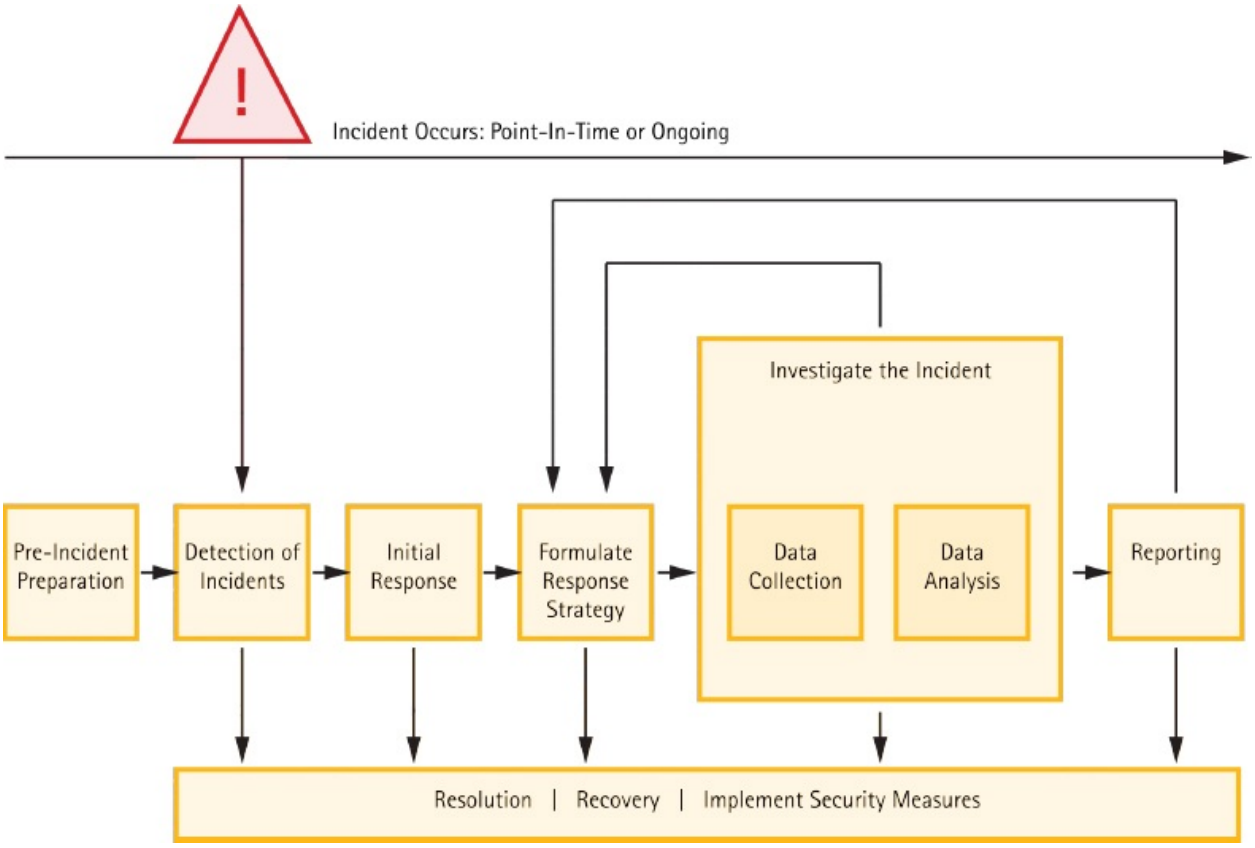


Abbildung 5.2: The Incident Response Process (Mandia u.a., 2003).

Phase 1: Pre-Incident Preparation

Die erste Phase umfasst die Vorbereitung der Organisation auf einen möglichen zukünftigen Vorfall. Die ***Pre-Incident-Preparation-Phase*** ist die einzige Phase, die ***vor*** dem eigentlichen Vorfall stattfindet. Beispielsweise kann sie die Installation von Sicherheitsmaßnahmen (IDS, Firewall etc.), die Einrichtung/Schaffung eines CSIRT (bzw. Definition, wie ein solches CSIRT bei Bedarf zusammengestellt wird), die Anschaffung von Hardware/Software für das CSIRT oder die Ausbildung der Mitglieder des CSIRT umfassen und stellt eine sogenannte ***Ongoing Phase*** dar.

Phase 2: Detection of Incidents

Die zweite Phase umfasst die eigentliche Erkennung des Sicherheitsvorfalls. Potentiell kann jeder einen Vorfall erkennen – vom nicht-technischen Personal bis hin zum Systemadministrator. Für diese Phase müssen Richtlinien existieren,

die beschreiben, wie auf verdächtige Ereignisse reagiert werden soll. Die Richtlinie sollte beispielsweise angeben, wer alarmiert werden soll und welche Daten weitergegeben und dokumentiert werden müssen. Die Richtlinie bildet die Grundlage für eine schnelle Reaktion des CSIRT.

Phase 3: Initial Response

In der ***Initial-Response-Phase*** werden weitere Daten über den Vorfall gesammelt. Hierzu können z.B. Interviews mit betroffenen Personen durchgeführt oder Protokolldateien gesichtet werden. Am Ende dieser Phase sollte das CSIRT in der Lage sein, folgende Fragen zu beantworten:

- Gab es überhaupt einen Sicherheitsvorfall?
- Falls ja, welcher Art ist der Vorfall?
- Wer und was ist von dem Vorfall betroffen?
- Was sind die Auswirkungen des Vorfalls für die Organisation?
- Wie hoch ist der mögliche Schaden?

Die Beantwortung dieser Fragen ist die Grundlage für die nächste Phase.

Phase 4: Formulation of Response Strategy

Fokus der vierten Phase ist die Formulierung einer Strategie, wie die Organisation mit dem Vorfall umgehen soll. Hierzu werden alle zur Verfügung stehenden Informationen aus der vorherigen Phase einbezogen („considering the totality of circumstances“ (Mandia u.a., 2003)). Jedoch muss die sogenannte ***response posture*** der Organisation, die Regeln angibt, wie mit bestimmten Vorfällen umzugehen ist, beachtet werden. Ein Beispiel einer solchen Regel könnte lauten: „Jeder Diebstahl muss zur Anzeige gebracht werden.“ Die Befolgung der ***response posture*** ist zentral für die Glaubwürdigkeit der Organisation.

Bei der Formulierung der Strategie ist meist die Beteiligung der verantwortlichen Personen, wie Management, Rechtsabteilung, Vorstand und Personalabteilung, nötig.

Phase 5: Investigation of the Incident

Während dieser Phase werden alle Spuren des Vorfalls untersucht, um die Fragen nach dem Was, Wann, Wie, Warum und Wer zu beantworten. Die Untersuchung des Vorfalls ist meist in zwei Unterphasen untergliedert:

- Sammeln von Daten (***data collection***)
- Analyse der Daten (***data analysis***)

Bei der Sammlung und Untersuchung von Spuren ***sollten*** Prinzipien der forensischen Informatik beachtet werden. Forensische Exaktheit ist jedoch oft nicht notwendig, weil in der Praxis häufig absehbar ist, dass keine straf- oder zivilrechtlichen Schritte eingeleitet werden.

Phase 6: Reporting

Das Reporting umfasst das Anfertigen eines Berichts zur Dokumentation aller Aktivitäten. Diese Dokumentation sollte sich auch an nicht-technisches Personal sowie Management und Vorstände richten und auch für diese Zielgruppe verständlich verfasst sein. Des Weiteren sollte der Bericht auch gerichtsverwertbar sein.

Phase 7: Resolution

Zielsetzung dieser letzten Phase ist es zu verhindern, dass der gleiche Vorfall sich wiederholt. Daher sollten Veränderungen am System erst nach Sicherung der Spuren vorgenommen werden. Dann sollten weitere Sicherheitsprozeduren installiert und deren Funktionsfähigkeit ständig überwacht werden. Vorhandene Sicherheitsprozeduren müssen, falls nötig, angepasst werden.

5.3 Der investigative Prozess

Der ***investigative Prozess*** von Casey (2004) stellt ein allgemeines Vorgehensmodell für digitale Untersuchungen dar, welches auch klassische Polizeiaufgaben einschließt, nicht nur die Aufgaben eines forensischen Experten.

Der investigative Prozess ist heute ein ***De-facto***-Standard in den angelsächsischen Ländern.

Der Prozess besteht aus insgesamt elf Phasen, die als Treppe visualisiert werden, beginnend mit der Alarmierung auf der untersten Stufe, bis hin zur Präsentation vor Gericht am Ende der Treppe (siehe Abbildung 5.3). Im Folgenden betrachten wir alle Phasen im Einzelnen und beziehen uns dabei auf die Übersetzung der einzelnen Begriffe nach Dornseif (2004). Abbildung 5.4 zeigt die deutschen Begriffe.

5.3.1 Anschuldigung

Die Anschuldigung ist das Startsignal für den gesamten Prozess. In dieser Phase werden zunächst die Quellen eingeschätzt und erste Erkundigungen eingeholt.

5.3.2 Güterabwägung

Im Rahmen der Güterabwägung wird das Interesse an der Verfolgung den Kosten, die bei der Verfolgung entstehen würden, gegenüber gestellt. Für Unternehmen fällt eine solche Abwägung (zumindest bei kleineren Vorfällen) meist gegen eine Verfolgung aus. Für eine Verfolgung sprechen neben der Chance auf Schadensersatz auch die Verbesserung der eigenen Sicherheit sowie eine gewisse Abschreckungswirkung. Gegen eine Verfolgung hingegen sprechen der Ressourcenverbrauch, unter Umständen die ***downtime***, in der die zu untersuchenden Systeme nicht produktiv eingesetzt werden können, und meist eine negative Öffentlichkeitswirkung.

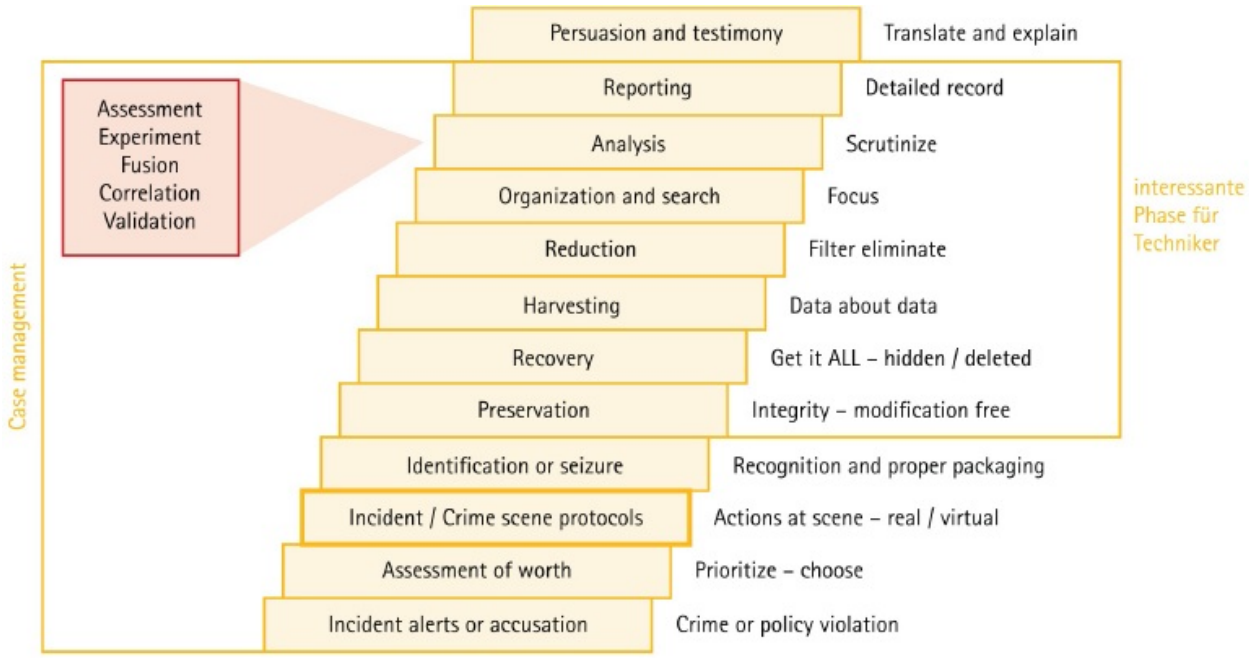


Abbildung 5.3: Der investigative Prozess (Casey, 2004).

5.3.3 Tatortsicherung

In der klassischen Kriminalistik wird immer gefordert, den Tatort weiträumig abzusperren. Casey (2004) formuliert dies wie folgt:

> „freeze the evidence in place and provide ground truth for all activities that follow.“

Für verschiedene Arten digitaler Spuren muss im Einzelfall überprüft werden, wie das Vorgehen des „Einfrierens“ genau aussieht. Insgesamt gilt es jeweils, die Gefahren der Verfälschung von Spuren so weit wie möglich einzudämmen.

Abbildung 5.4: Der investigative Prozess in der Übersetzung von Dornseif (2004).

5.3.4 Beschlagnahme

Bei einer traditionellen Beschlagnahme werden alle Gegenstände mitgenommen, die als Beweismittel dienen könnten. Wichtig ist hierbei, nichts an den Beweismitteln zu verändern. Aber auch die Umgebung der gesicherten Beweismittel kann von großer Relevanz sein. Mit der Beschlagnahme beginnt die ***chain of custody***.

Exkurs 11 (Sicherstellung vs. Beschlagnahme) ***In der deutschen Strafprozessordnung (StPO) wird in*** § ***94 zwischen den Begriffen „Sicherstellung“ und „Beschlagnahme“ unterschieden. Unter einer Sicherstellung versteht man den*** freiwilligen Gewahrsamsübergang ***an die Ermittlungsbehörde. Im Gegensatz spricht man von Beschlagnahme, wenn die Beweismittel nicht freiwillig herausgegeben werden und trotzdem sichergestellt werden sollen***.

Eine gute Lektüre zur Durchführung von Beschlagnahmen ist die vom US Department of Justice/National Institute of Justice herausgegebene Broschüre „Electronic Crime“ Scene Investigation: A Guide to First Responders

(Department of Justice u. National Institute of Justice, 2001). Sie gibt ausführliche und genaue Hinweise auch für nichttechnisches Personal, wie die Beispiele in den Abbildungen 5.5 bis 5.9 zeigen.

Eine weitere gute Quelle ist das Dokument „Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations“ vom United States Department of Justice (Department of Justice, 2002).

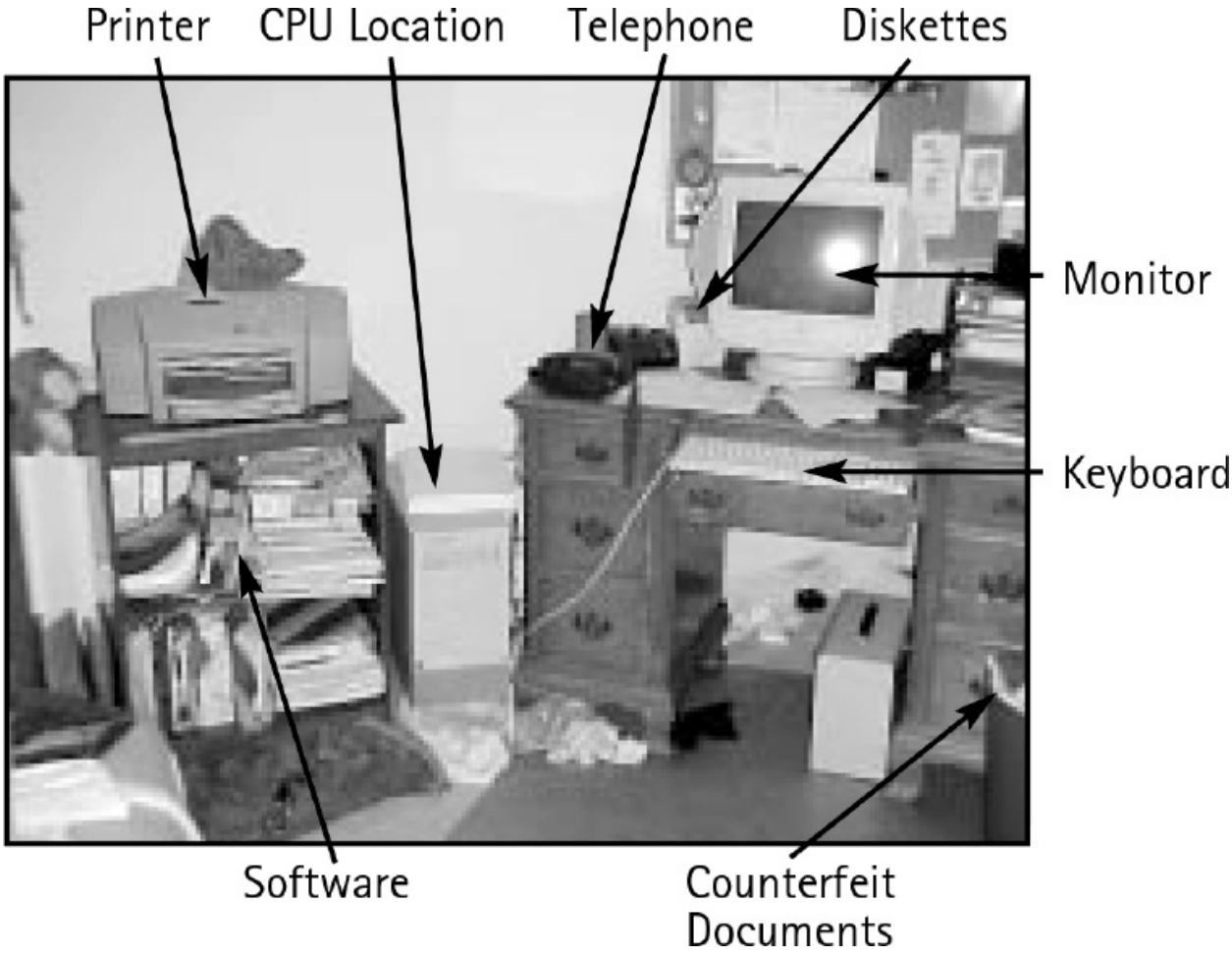


Abbildung 5.5: Arten digitaler Geräte am Tatort (Department of Justice u. National Institute of Justice, 2001).

5.3.5 Sicherung

Bei der Sicherung der Beweismittel gilt es sicherzustellen, dass diese unverändert bleiben. Daher werden alle Beweismittel dokumentiert, fotografiert, versiegelt und anschließend weggeschlossen. Bei digitalen Spuren bedeutet das

in der Regel, zuerst Kopien der Beweismittel zu erstellen und weitere Untersuchungen nur auf den Kopien durchzuführen. Zum Nachweis der Echtheit von Beweismittelkopien kommen kryptographische Hashfunktionen zum Einsatz. Außerdem ist es wichtig, ausschließlich vertrauenswürdige Software zu gebrauchen. Bei der Sicherungsphase beginnt die Arbeit von Informatik-Spezialisten.

Abbildung 5.6: Unterschiedliche Arten von Kabeln (Department of Justice u. National Institute of Justice, 2001).

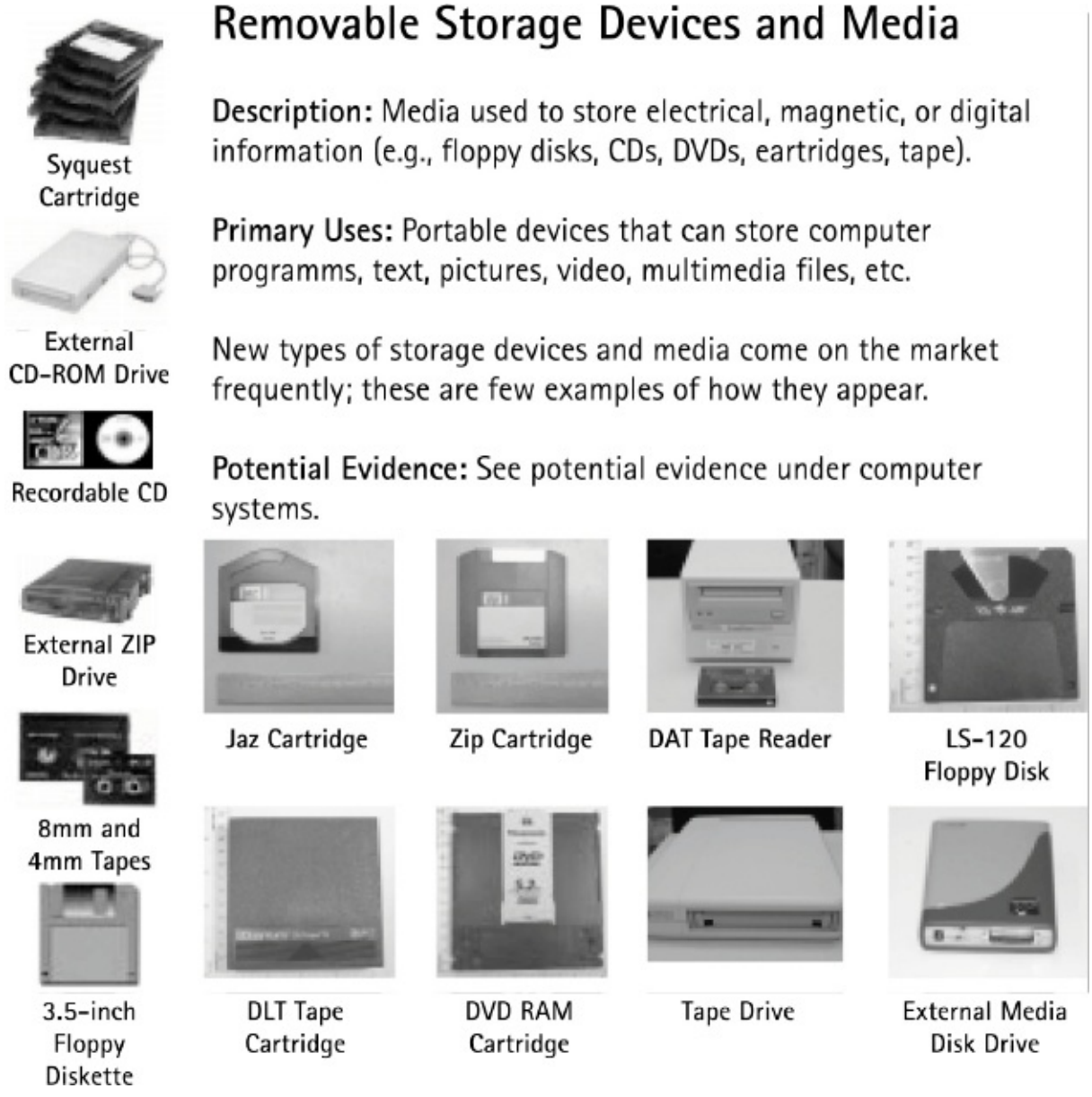


Abbildung 5.7: Beschreibung von Speichermedien (Department of Justice u. National Institute of Justice, 2001).

Answering Machine

Answering Machines

Description: An electronic device that is part of a telephone or connected between a telephone and the landline connection.
Some models use a magnetic tape or tapes, while others use an electronic (digital) recording system.

Primary Uses: Records voice messages from callers when the called party is unavailable or choose not to answer a telephone call. Usually plays a message from the called party before recording the message.

Note: Since batteries have a limited life, data could be lost if they fail. Therefore, appropriate personnel (eg., evidence custodian, lab chief, forensic examiner) should be informed that a device powered by batteries is in need of immediate attention.

Potential Evidence: Answering machines can store voice messages and, in some cases, time and date information about when the message was left. they may also contain other voice recordings.

- Caller indentification information
- Deleted messages
- Last number called
- Memo
- Phone numbers and names
- Tapes

Abbildung 5.8: Sonstige elektronische Spuren (Department of Justice u. National Institute of Justice, 2001).

Packaging procedure:

a. Ensure that all collected alectronic evidence is properly documented labeled, and inventoried before packaging.

b. Pay special attention to latent or trace evidence and take actions to preserve it.

c. Pack magnetic media in antistatic packaging (paper or antistatic plastic bags).
Avoid using materials that can produce static electricity, such as standard plastic bags.

d. Avoid folding, bending, or scratching computer media such as diskettes, CD-ROMs, and tapes.

e. Ensure that all containers used to hold evidence are properly tabeled

Abbildung 5.9: Beispiel für die Dokumentation der vorgefundenen Verkabelung (Schreibfehler in der ersten Zeile entstammt der Quelle) (Department of Justice u. National Institute of Justice, 2001).

5.3.6 Bergung

Casey (2004) beschreibt die Bergung als „***throwing out a large net***“. Insbesondere umfasst diese Phase die Bergung von Daten, die gelöscht, versteckt, getarnt oder anderweitig unzugänglich gemacht wurden. Hier empfiehlt es sich, Synergien mit anderen Beweismitteln zu nutzen. Beispielsweise ist es sinnvoll zu prüfen, ob ein Zettel mit Passworten am Tatort gefunden wurde, wenn verschlüsselte Daten gelesen werden müssen.

5.3.7 Auswertung

Bei der Auswertung ist vor allem eine gute Organisation der üblicherweise großen Datenmenge erforderlich. Hierzu sollte zunächst eine Untersuchung von

Metadaten anstatt der eigentlichen Daten vorgenommen werden. Beispielsweise können Daten nach Dateityp oder Zugriffszeiten gruppiert werden. Dies führt direkt zur nächsten Phase, der Reduktion.

5.3.8 Reduktion

Aufgabe der Reduktion ist es, irrelevante Daten zu eliminieren. Hierzu kann auch weiterhin auf Metadaten gearbeitet werden. Zum Beispiel können die Daten aufgrund des Datentyps reduziert werden. Ein Szenario hierfür wäre bei entsprechender Anschuldigung die Reduktion aller Dateien auf Bilddateien. Das Ergebnis dieser Phase ist nach Casey ***„ the smallest set of digital information that has the highest potential of containing data of probative value“*** , also die kleinste Menge digitaler Information mit der größten Wahrscheinlichkeit, beweiskräftige Daten zu enthalten. Hilfreich sind hier Hash-Datenbanken von bekannten Dateien, wie zum Beispiel ***The NIST National Software Reference Library*** (NIST, 2011), um bereits bekannte Dateien ausschließen zu können.

Weitere Teilaspekte der Reduktion sind die Strukturierung der Daten sowie die Durchsuchbarmachung, um die Daten nach der Reduktion zu organisieren. Hierzu werden häufig Indizes und Übersichten erstellt. Dies vereinfacht die Referenzierung der Daten in den nachfolgenden Schritten.

5.3.9 Analyse

Diese Phase beinhaltet die Detailanalyse unter Beachtung der Dateiinhalte. Unter anderem müssen Verbindungen zwischen Daten und Personen hergestellt werden, um Verantwortliche zu ermitteln. Weiterhin erfolgt die Bewertung von Inhalt und Kontext nach Bedeutung, Motivation und Gelegenheit (means, ***motivation, opportunity***). Experimente sind in dieser Phase hilfreich, um undokumentiertes Verhalten zu ermitteln und neue Methoden zu entwickeln. Alle Ergebnisse müssen durch wissenschaftliche Methodik überprüft werden und überprüfbar sein.

5.3.10 Bericht

Aufgabe des Berichtes ist es nicht ausschließlich Ergebnisse zu präsentieren, sondern auch darzulegen, wie diese erlangt wurden. Hierzu sollten immer auch die befolgten Regeln und Standards im Bericht dokumentiert werden. Alle

gezogenen Schlüsse müssen begründet und auch alternative Erklärungsmodelle erörtert werden.

5.3.11 Bezeugen

Schlussendlich kommt es zur Bezeugung als Sachverständiger vor Gericht. Wichtigster Punkt ist die Glaubwürdigkeit des Bezeugenden. Problematisch können hierbei ein technikfeindliches Publikum oder problematische, beispielsweise vom Verteidiger angeführte, Analogien sein.

5.4 Das Common Model

Das ***Incident-Response***-Modell aus Abschnitt 5.2 legt den Fokus auf das Management eines Sicherheitsvorfalls und die Integration der Untersuchung in die Prozesse einer Organisation. Oftmals steht auch die schnelle Wiederherstellung der Produktionssysteme im Vordergrund. Der investigative Prozess aus Abschnitt 5.3 hingegen legt den Fokus auf ein exaktes Vorgehen bei der Beweissicherung, die hierzu in viele Phasen aufgeteilt ist, und auf die gerichtsverwertbare Sicherung der Spuren mittels wissenschaftlicher Methodik.

Offenbar ergänzen sich diese beiden Modelle. Um dies zu verdeutlichen, destillierten Freiling u. Schwittay (2007) den kleinsten gemeinsamen Nenner beider Modelle in ein vereinigendes Modell, das sogenannte ***Common Model***. Das Wort ***common*** bezieht sich dabei auf die Tatsache, dass es die Gemeinsamkeiten zweier Extreme verdeutlichen soll. Es ist also eher ein Versuch, die Gemeinsamkeiten aller Modelle herauszuarbeiten als ein ernstgemeinter Versuch, ein praktikables Modell für den alltäglichen Gebrauch zu entwickeln. Wir geben im Folgenden einen kurzen Überblick über das Common Model.

5.4.1 Überblick

Das Common Model ist ein abstraktes, gemeinsames Vorgehensmodell für Incident Response und Computerforensik. Das Modell stellt eine Ergänzung des investigativen Prozesses um eine Management-Komponente, beziehungsweise eine Ergänzung des ***Incident Response***-Modells um eine detaillierte Analysekomponente dar. Das Common Model ist noch abstrakter als die beiden

Ausgangsmodelle und somit eher als didaktischer Rahmen denn als konkretes Vorgehensmodell für die Praxis zu verstehen. Da es auf die beiden Ausgangsmodelle aufbaut, wirkt es sicherlich in der folgenden Beschreibung etwas repetitiv, was didaktisch beabsichtigt ist.

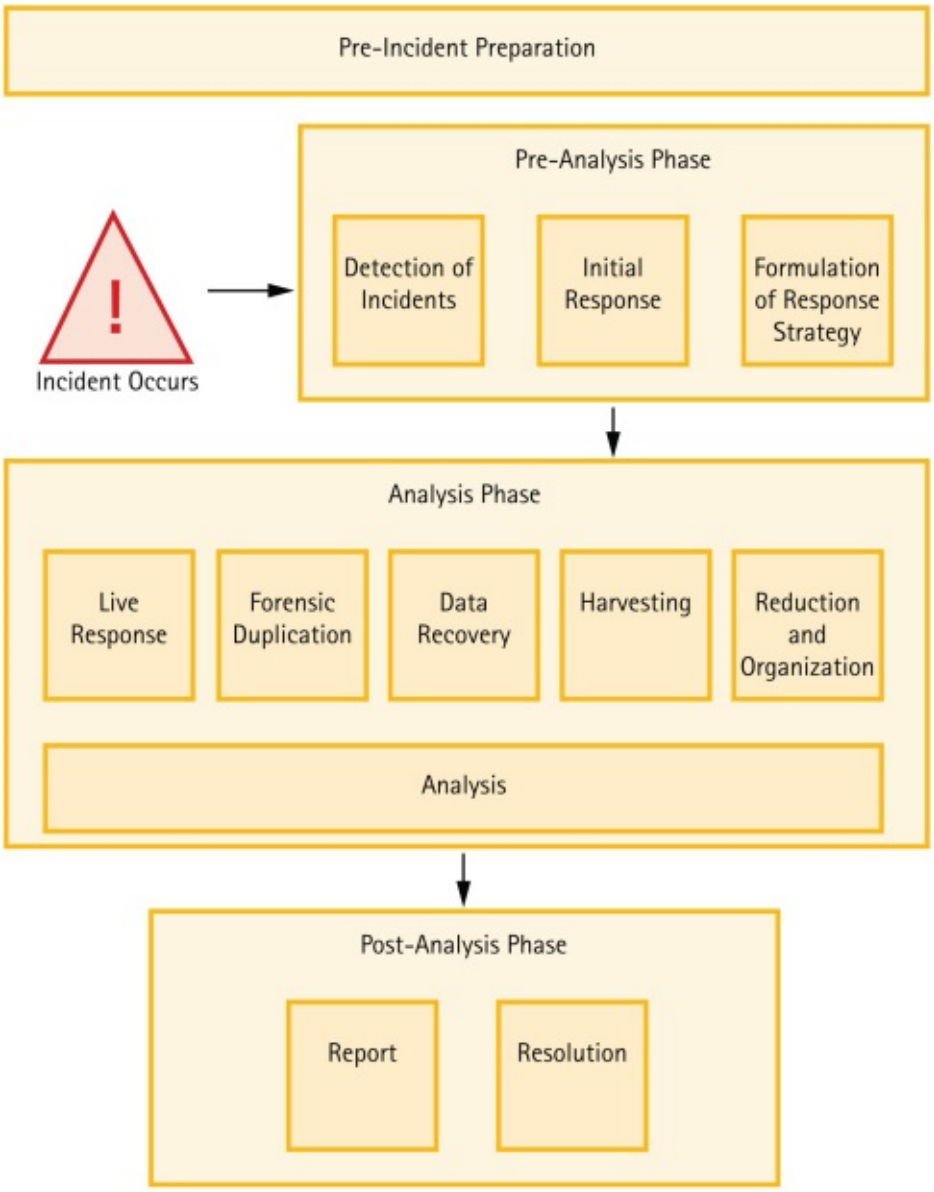


Abbildung 5.10: Überblick über das Common Model

Das Common Model umfasst die folgenden drei Phasen, die jeweils in mehrere Schritte untergliedert sind (siehe Abbildung 5.10):

1. Pre-Analysis Phase

2. Analysis Phase
3. Post-Analysis Phase

5.4.2 Pre-incident Preparation

Ziel der Pre-Incident Preparation ist es, ein Unternehmen oder eine Einrichtung in die Lage zu versetzen, einen Vorfall in einer wohldefinierten Weise zu behandeln, um eine schnelle und effektive Lösung zu ermöglichen. Die üblicherweise während der Pre-Incident Preparation Phase unternommenen Schritte lassen sich in zwei Hauptkategorien unterteilen:

- Schritte, die das zuständige Personal in speziellen CSIRTs auf zukünftige Vorfälle vorbereiten,
- und solche, welche die Organisation selbst auf Vorfälle vorbereiten.

Ein besonderer Aspekt in Bezug auf die Vorbereitung einer Organisation auf zukünftige Vorfälle ist die Ausarbeitung einer Policy, die definiert, wie in der Organisation auf bestimmte Arten von Vorfällen reagiert werden soll. Beispielsweise stellt sich die Frage, in welchem Umfang der Netzwerkverkehr überwacht werden soll. Hierbei muss sichergestellt werden, dass die Durchführung einer Untersuchung gewährleistet ist, ohne die Privatsphäre der Mitarbeiter zu verletzen oder gegen Datenschutzvereinbarungen oder öffentliches Recht zu verstoßen.

5.4.3 Pre-Analysis Phase

Die Pre-Analysis Phase umfasst alle Schritte, die zwischen dem Erkennen eines Vorfalls und dessen eigentlicher Analyse liegen. Ihre einzelnen Schritte sind in Abbildung 5.11 dargestellt.

Incident Detection

Die Incident Detection befasst sich mit der Aufstellung von Richtlinien zur schnellen Erkennung von Vorfällen. Außerdem müssen Kommunikationswege für die Benachrichtigung über Vorfälle sowie die Art und Weise der Berichterstattung festgelegt werden. Hierdurch soll eine möglichst rasche

Übergabe des Vorfalls an das zuständige CSIRT ermöglicht werden. Incident Detection findet normalerweise statt, sobald eine Person oder ein Sicherheitsmechanismus eine unautorisierte oder ungesetzmäßige Aktivität vermutet. Tatsächlich gibt es eine ganze Reihe von Quellen, aus der solche Vermutungen kommen können, wie beispielsweise Angestellte, Sicherheitspersonal, Intrusion-Detection-Systeme oder Systemadministratoren. Aus diesem Grund ist es auch wichtig, dass Richtlinien genau bestimmen, wer bei einem Vorfall zu informieren und wie in einer solchen Situation zu reagieren ist. Dies ist unter anderem auch notwendig, um möglichst wenige potentielle Beweismittel zu zerstören.

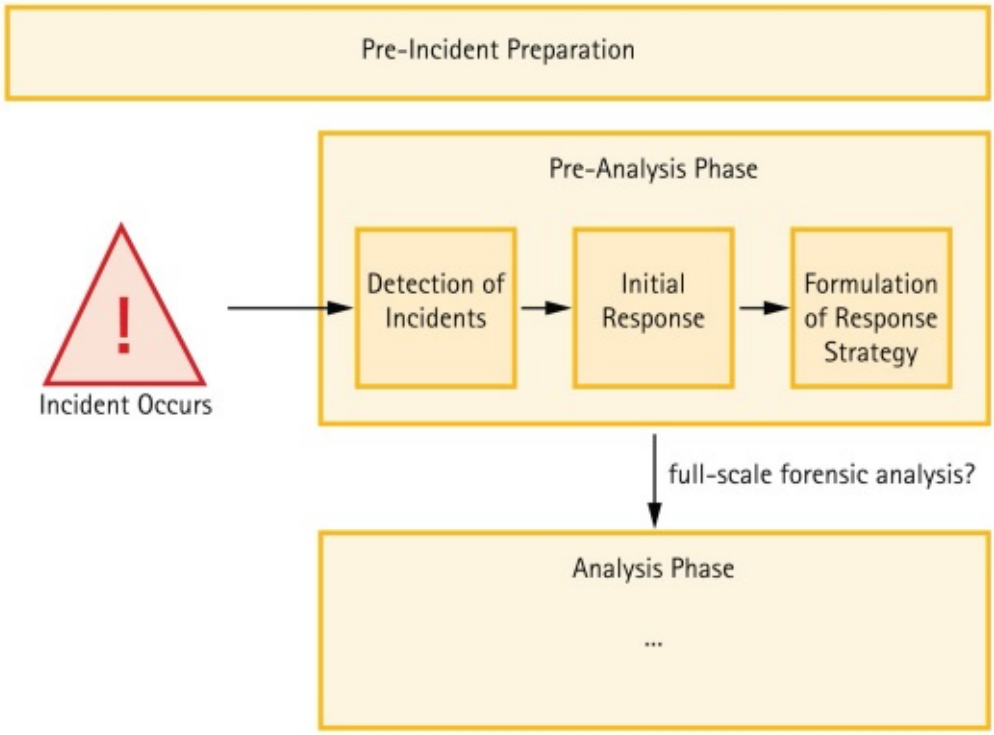


Abbildung 5.11: Common Model: Pre-Analysis Phase

Wenn ein Vorfall gemeldet wird, sollten zunächst einige grundlegende Informationen, wie Datum und Uhrzeit der Erkennung des Vorfalls sowie die betroffenen Personen und Systeme, aufgenommen werden. Im Idealfall sollten vorgefertigte Formulare zur Verfügung gestellt werden, um eine standardisierte Dokumentation zu gewährleisten. Im Anschluss an die Aufnahme sollte das CSIRT alarmiert und informiert werden, so dass dieses den Vorfall zur weiteren Untersuchung übernehmen kann.

Initial Response

Das Ziel der Initial Response ist es, den Verdacht auf einen Vorfall durch das CSIRT entweder zu bestätigen oder zu verwerfen. Wenn der Vorfall bestätigt wurde, sollten Art und Umfang des Vorfalls festgestellt werden, um eine angemessene Strategie zur Verfolgung des Vorfalls in den folgenden Schritten festlegen zu können. Initial Response beinhaltet aber auch das Ergreifen geeigneter Maßnahmen, um den Vorfall einzudämmen und den potentiellen Schaden zu begrenzen. Häufig wird zur Beobachtung eines laufenden Vorfalls die Überwachung des Netzwerkverkehrs eingeleitet. In einigen wenigen Fällen kann Initial Response sogar Schritte beinhalten, die normalerweise erst während der Live Response ergriffen werden. In diesem Fall müssen alle Voraussetzungen, die für Live Response gelten, auch in diesem frühen Schritt bereits berücksichtigt werden. Möglicherweise kann auch weiterer Schaden für die Organisation vermieden werden, indem betroffene Systeme oder verdächtige Personen von anderen Systemen oder dem „Tatort“ getrennt werden. Da ohnehin immer alle Aktivitäten dokumentiert werden müssen, werden alle gewonnenen Information gesammelt sobald der Vorfall bestätigt wurde. Weiterhin werden alle gewonnenen Daten genutzt, um die Auswirkungen des Vorfalls auf Benutzer, Systeme und den Produktionsbetrieb abzuschätzen. Diese Einschätzung bildet die Grundlage für die Formulierung einer adäquaten Strategie zur Behandlung des Vorfalls im nächsten Schritt.

Formulation of Response Strategy

Ziel dieses Schrittes ist es, die beste Strategie zur Behandlung des konkreten Vorfalls zu finden. Insbesondere muss das CSIRT entscheiden, ob eine vollständige forensische Analyse des Vorfalls durchgeführt werden soll. Es gibt eine Menge Faktoren, die die Entscheidungsfindung beeinflussen. Beispiele für offensichtliche Eigenschaften eines konkreten Vorfalls, die Einfluss auf die angewandte Strategie haben, sind die Einschätzung, wie kritisch das kompromittierte System für die Organisation ist, eine Bewertung der Fähigkeiten des Angreifers oder auch mögliche finanzielle Verluste. Diese Faktoren beeinflussen maßgeblich, wie viele Ressourcen zur Untersuchung des Vorfalls aufgebracht werden sollen. Weitere Faktoren, die einen großen Einfluss auf die Wahl der Strategie haben können, sind beispielsweise politische Entscheidungen (etwa die Frage, was geschieht, wenn der Vorfall öffentlich bekannt wird),

rechtliche Beschränkungen (zum Beispiel Haftbarkeit für das Unterlassen einer Meldung des Vorfalls an die Strafverfolgungsbehörden) oder geschäftliche Ziele.

5.4.4 Analysis Phase

Während der Analysis Phase findet die eigentliche Analyse der kompromittierten Systeme statt (siehe Abbildung 5.12). Den Umfang der Analyse bestimmt die im vorherigen Schritt entwickelte Strategie. Zunächst werden durch Live Response auf dem betreffenden System noch im Betrieb Daten gesichert. Der Rest der Analysis Phase beschäftigt sich mit der Post-Mortem-Analyse der Systeme. Falls in der vorherigen Phase die Entscheidung zugunsten einer vollständigen forensischen Analyse fiel, müssen alle Schritte der Analysis Phase ohne Ausnahme durchgeführt werden. Wenn eine solch gründliche Untersuchung nicht gewünscht ist, können manche Schritte ausgelassen oder verkürzt werden – dies wird in den folgenden Schritten jeweils konkret angemerkt. Bestimmte Strategien können sogar eine Analyse gänzlich überflüssig machen und das CSIRT kann direkt zur Resolution Phase übergehen.

Live Response

Live Response beinhaltet die Datensammlung auf laufenden Systemen. Das bedeutet, dass die zu analysierenden Systeme angeschaltet bleiben und im laufenden Betrieb analysiert werden. Ziel der Live Response ist es, flüchtige Daten zu sichern, die nach Abschalten des Systems verloren gingen. Zusätzlich ist es häufig Praxis, auch einige nichtflüchtige (persistente) Daten zu sichern. Alle durchgeführten Schritte sollten hierbei das System so wenig wie möglich verändern, um das Originalbeweismittel nicht mehr als unbedingt nötig zu modifizieren. Bestimmte Daten, wie etwa offene Netzwerkverbindungen und laufende Prozesse, werden ausschließlich im Hauptspeicher gespeichert und gehen daher schnell verloren, wenn das System ausgeschaltet wird. Wird der Hauptspeicherinhalt nicht gesichert, besteht auch keine Möglichkeit, diese Daten später zu rekonstruieren. Die Kombination aller während der Live Response gesammelten Daten stellt oft eine wichtige Information für den Ermittler dar, um das Ausmaß eines Vorfalls abzuschätzen und geeignete Maßnahmen zu ergreifen.

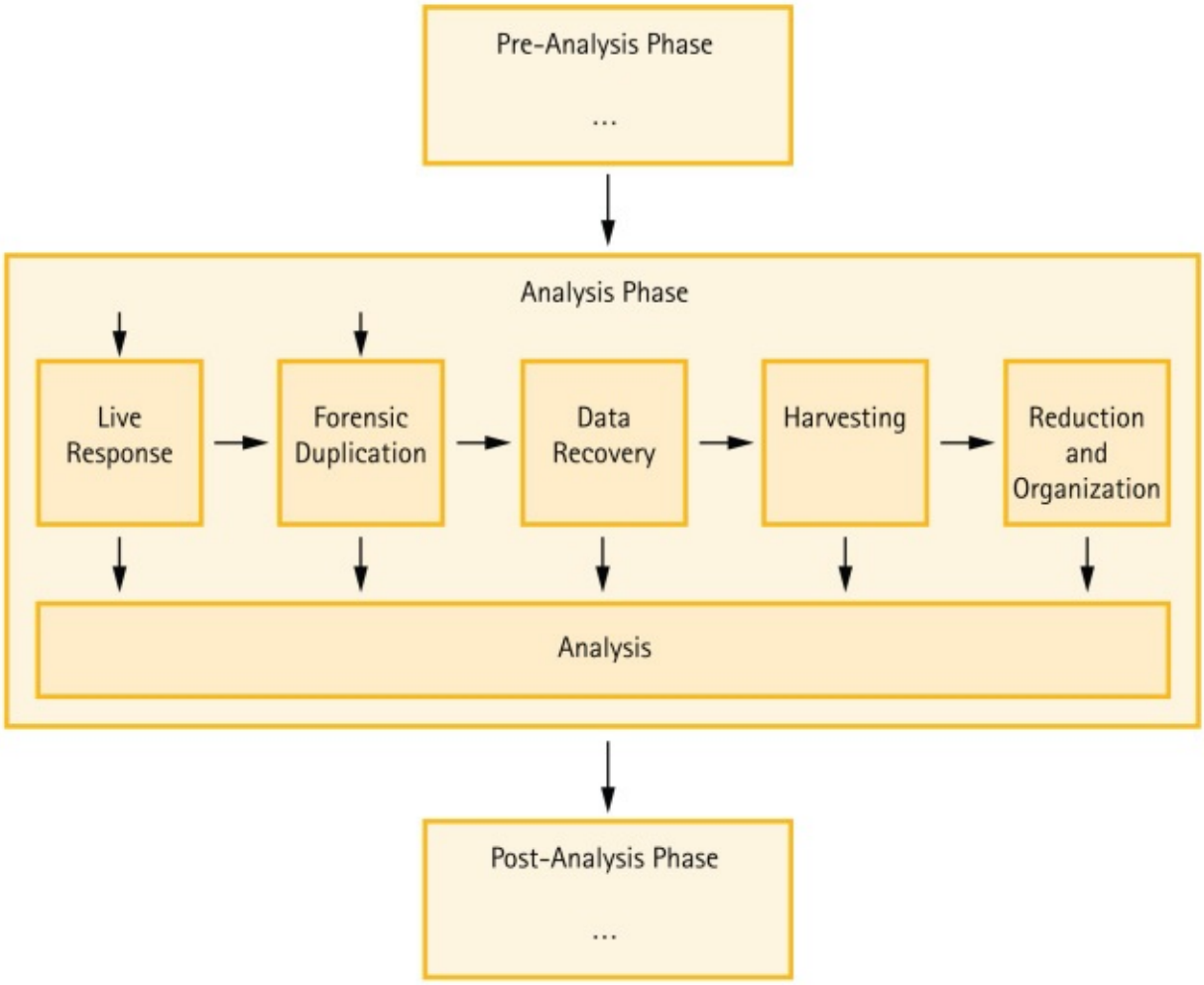


Abbildung 5.12: Common Model: Analysis Phase

Forensische Duplizierung

In der forensischen Duplizierung (***imaging***) werden exakte Kopien aller Speichermedien, die in Zusammenhang mit dem Vorfall stehen, angefertigt. Die Originalbeweismittel müssen hierbei unverändert bleiben. Eine ***chain of custody*** wird für alle Beweismittel angelegt, und das Originalbeweismittel wird zusammen mit einem Duplikat an einem sicheren Ort (Asservatenkammer) eingelagert.

Data Recovery

Dieser Schritt befasst sich mit der Wiederherstellung von Daten, die im aktuellen Zustand eines im vorherigen Schritt erstellten Duplikates nicht ohne Weiteres zur Analyse zur Verfügung stehen. Data Recovery beinhaltet das Wiederherstellen von gelöschten, beschädigten, versteckten oder auf andere Art und Weise unzugänglichen Daten in einem Dateisystem oder in unallozierten Bereichen des Datenträgers (siehe Carrier (2005)).

Harvesting

Während des Harvesting beginnt der Ermittler, Metadaten (Daten über Daten, wie etwa Zeitstempel oder Dateigrößen) über die im vorherigen Schritt erhobenen Daten zu sammeln. Ziel ist es hierbei, das größtenteils unorganisierte Material nach bestimmten Kriterien, wie Zeitstempeln oder anderen Metadaten zu strukturieren. Diese Struktur kann für die weiteren Ermittlungen nützlich sein, da häufig bestimmte Eigenschaften auf einen Zusammenhang mit dem Vorfall hindeuten, wie etwa alle Dateien, die in einem bestimmten Zeitraum modifiziert wurden.

Bei der Untersuchung von Dateien kann die Information der Dateitypen genutzt werden, um alle Dateien eines bestimmten Typs zu gruppieren. Beispielsweise kann für einen konkreten Vorfall lediglich der Besitz von bestimmtem Video- und Bildmaterial von Bedeutung sein. Hingegen können bei einem Verdacht auf ein Rootkit ausführbare Dateien und Treiber im Fokus der Ermittlung stehen.

Reduction and Organization

Während der Reduktion und Organisation können alle Daten, die als irrelevant für den Vorfall identifiziert wurden, ausgeschlossen werden. Ziel dieses Schrittes ist es, die große Menge von strukturierten Daten, die das Ergebnis des vorherigen Schrittes war, auf die kleinste Menge digitaler Information zu reduzieren, die das größte Potential aufweist, beweiskräftige Daten zu enthalten. Die verbleibenden Daten werden organisiert, um den späteren Zugriff zu erleichtern, indem beispielsweise das effiziente Suchen und die Identifikation und Referenzierung relevanter Daten in den folgenden Schritten, etwa durch Nummerierung und Bezeichnung, ermöglicht werden.

Analysis

Nachdem in den vorherigen Schritten alle Daten mit Bezug zu dem konkreten Vorfall wiederhergestellt, gesammelt, reduziert und organisiert wurden, beginnt in diesem Schritt die eigentliche Analyse. Ein Ermittler entwickelt Hypothesen darüber, was geschehen ist, wann und wie es geschehen ist und wer dafür verantwortlich ist. Zumindest bei einer Ermittlung wegen einer kriminellen Handlung muss durch die Ermittlung im Idealfall ein Täter identifiziert werden. Außerdem müssen die Fragen nach den Hintergründen, dem Motiv und der Gelegenheit beantwortet werden. Beim Sichten des tatsächlichen Inhalts der gesammelten Daten werden verschiedene digitale Beweisstücke in Bezug zueinander gesetzt und Verbindungen hergestellt. Durch vollständige Dokumentation aller Aktivitäten und Überprüfung der Ergebnisse resultiert dieser Schritt in einer vollständigen Rekonstruktion des Vorfalls, basierend auf objektiven und wissenschaftlichen Prinzipien.

Um eine objektive Analyse und Interpretation der Ergebnisse zu gewährleisten, müssen wissenschafltiche Methoden angewandt werden. Das bedeutet, dass ein Ermittler wie bereits erläutert unterschiedliche Hypothesen bezüglich eines Vorfalls aufstellen und versuchen muss, sie zu widerlegen – statt sie zu beweisen. Durch das Ausschließen von Theorien, die mit den bereits gewonnenen Erkenntnissen nicht vereinbar sind, ist es schwieriger, von einer Hypothese eingenommen zu werden, und die verbleibenden Hypothesen haben eine höhere Wahrscheinlichkeit, eine korrekte Rekonstruktion des Hergangs darzustellen.

Eine weitere wichtige Eigenschaft der Ergebnisse der Analysis Phase ist die Wiederholbarkeit. Das bedeutet, jeder Beobachter muss die gleichen Beobachtungen wie der Ermittler machen können, indem er die gleichen Methoden anwendet. Hierzu werden während der Analyse jeder unternommene Schritt, jede angewandte Technik und jedes benutzte Werkzeug sofort präzise dokumentiert. Ebenso werden die resultierenden Ergebnisse und deren Auswirkung auf die in Betracht kommenden Hypothesen schriftlich festgehalten.

5.4.5 Post-Analysis Phase

Die Post-Analysis Phase beginnt, nachdem alle Aktivitäten zur Sammlung und Analyse digitaler Beweismittel abgeschlossen sind und die in der Strategie

formulierten Ziele der Analysis Phase erreicht wurden. Einen Überblick über diese Phase gibt Abbildung 5.13.

Report

Dieser Schritt beinhaltet das Verfassen eines präzisen Berichtes, der die Details eines Vorfalls in einer, auch für nicht-technische Leser, verständlichen Art und Weise beschreibt. Der Bericht enthält die gesamte Dokumentation, die während der Pre-Analysis und Analysis Phase erstellt wurde, und kombiniert diese zu einer verständlichen Übersicht über den gesamten Fall. Hierbei werden die wichtigsten Ergebnisse der Analyse und deren Bedeutung für die Lösung des Vorfalls aufgezeigt.

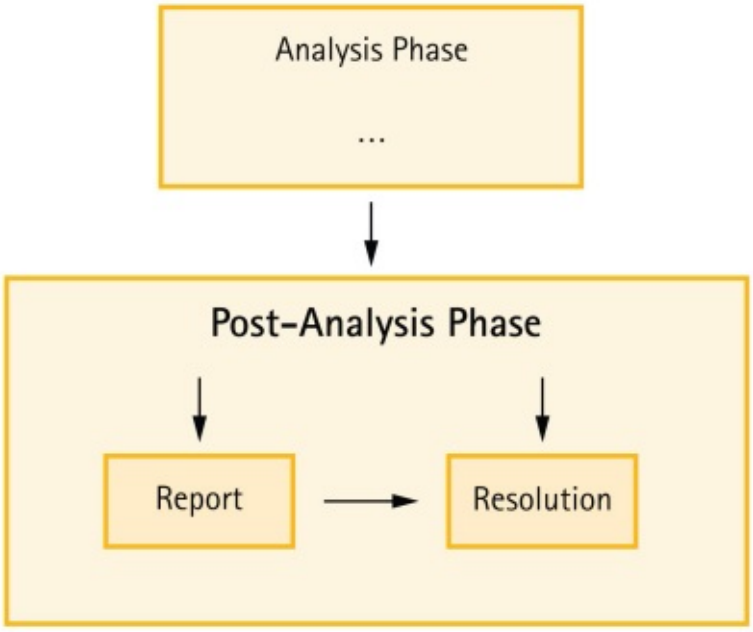


Abbildung 5.13: Common Model: Post-Analysis Phase.

Resolution

Ziel der Resolution ist es, das Problem, welches den Vorfall verursachte, zu erkennen, es zu lösen und geeignete Maßnahmen zu treffen, um einen erneuten Vorfall dieser Art in Zukunft zu verhindern. Im Falle eines laufenden Vorfalls sollten diese Maßnahmen bereits vor dem Abschluss der Analyse durchgeführt werden, jedoch (wenn möglich) erst nachdem alle potentiellen Beweismittel

gesichert wurden. Um zu überprüfen, dass die getroffenen Sicherheitsmaßnahmen erfolgreich umgesetzt wurden und in der geplanten Art und Weise funktionieren, sollte die Umsetzung überwacht und die Effektivität der Maßnahme getestet werden.

5.4.6 Diskussion

Das Common Model für Incident Response und Computerforensik zeigt einen Weg auf, um Incident Response durchzuführen, indem aus der Computerforensik bekannte Prinzipien während der Analysis Phase angewandt werden. Dies soll die Nachteile der beiden einzelnen Modelle ausgleichen. Das Modell nutzt die Management-Prozeduren des Incident-Response-Modells, welches maßgeblich die Pre-Analysis und die Post-Analysis Phase beschreibt. Üblicherweise werden Maßnahmen wie Pre-Incident Preparation und Incident Detection nicht als Teile der herkömmlichen investigativen Vorgehensmodelle betrachtet, da sie das Problem aus der Perspektive der Strafverfolgungsbehörden sehen.

Zunächst ist festzustellen, dass unterschiedliche Anforderungen generell unterschiedliche Vorgehensweisen verlangen. Falls der vermutete Angreifer eine große Bedrohung darstellt, sollte man vorsichtiger (also forensisch genauer) vorgehen. Wenn hohe Schäden erwartet werden, sollte man auch forensisch genau vorgehen (Schadensersatz). Im Prozess sollte man daher einen Punkt vorsehen, an dem das genaue weitere Vorgehen entschieden wird: Soll eine vollständige forensische Untersuchung stattfinden oder nicht? Ein formales Kriterium für diese Entscheidung könnte etwa die Frage sein, ob das Bedrohungspotential multipliziert mit dem möglichen Schaden größer als eine bestimmte festgelegte Schranke ist.

Eine Randbedingung für die Anwendung der erläuterten Vorgehensmodelle ist die sogenannte ***response posture***, also die Frage, wie die Organisation generell mit Vorfällen einer bestimmten Klasse umgeht. Verfolgt die Organisation beispielsweise eine „Null-Toleranz-Strategie“, erübrigt sich die Entscheidung, ob der Vorfall verfolgt werden soll. Des Weiteren müssen rechtliche Bestimmungen in Betracht gezogen werden. Unter Umständen kann zum Beispiel die Auslassung einer Reaktion selbst wiederum strafbar sein.

Ein weiteres Problem insbesondere der digitalen Forensik ist, dass das Prinzip „***throwing out a large net***“ zunehmend problematisch wird. Ermittler ersticken

immer häufiger in Datenbergen. Eine wichtige Frage aktueller Forschung ist daher, wie viel Reduktion bereits vor der Bergung möglich ist.

5.5 Weitere Modelle

Wir geben abschließend einen kurzen Überblick über weitere in der Literatur vorgeschlagenen Modelle. Eine Beschreibung weiterer forensischer Modelle liefert der Artikel von Pollitt (2007). Die Vorstellung der Modelle erfolgt chronologisch.

Pollitt 1995

Pollitt (1995) stellte bereits 1995 eines der ersten Vorgehensmodelle vor. Er stellt in seiner Arbeit fest, dass das Gesetz in der analogen Welt einen Prozess voraussetzt, der von allen beteiligten Parteien verstanden wird und beobachtet werden kann. Dieser Prozess besteht aus vier Schritten, die in der Regel vollkommen unbewusst stattfinden. Bei dem Versuch, diesen Prozess auf digitale Beweismittel anzuwenden, ergeben sich jedoch neue Probleme.

Zunächst wird ein Dokument sichergestellt. Die Art und Weise der Sicherstellung wird durch eine Menge althergebrachter und gut dokumentierter Regeln festgelegt. Festzustellen, ob ein Dokument physikalisch existiert oder woher es stammt, ist nur in seltenen Fällen ein Problem – ganz im Gegensatz zu digitalen Beweismitteln.

Als nächster Schritt wird das sichergestellte Dokument einem Identifikationsprozess unterzogen. Ist das Dokument beispielsweise in englischer Sprache verfasst, sollte es in der Regel einer der englischen Sprache mächtigen Person möglich sein festzustellen, was der Inhalt des Dokumentes ist. Eine Binärdatei hingegen erfordert zunächst eine Übersetzung durch ein Programm, das die Daten in eine für Menschen lesbare Form überführt. Erst dann ist eine Person in der Lage festzustellen, welchen Inhalt das Dokument hat.

Danach kann die Auswertung des Dokumentes erfolgen, und der Ermittler entscheidet, ob das Dokument relevant ist und wer eine Aussage bezüglich des Dokumentes machen könnte. Wenn digitale Beweismittel in eine lesbare Form gebracht sind, können diese Entscheidungen ebenfalls gefällt werden. Jedoch ist

der elektronische Kontext einer Datei noch immer entscheidend. Dies hat Auswirkungen darauf, wie und durch wen das Beweismittel vorgebracht wird.

Schließlich wird das Dokument als Beweis vorgebracht. Dies kann ausschließlich durch eine Person erfolgen, die in der Lage ist, seine Herkunft und Bedeutung zu erläutern. Im Falle eines nicht-digitalen Beweismittels können Richter und Anwälte das Dokument physisch in Augenschein nehmen und anschließend die Aussage des Sachverständigen anhören. Es ist hierbei nicht notwendig, auf die vorhergehenden Schritte einzugehen, da diese generell akzeptiert sind. Dies gilt jedoch nicht für digitale Beweismittel. Es ist vielmehr häufig erforderlich, dass ein Sachverständiger den Prozess der Sicherstellung, Identifikation und Auswertung bezeugt.

Dieser Prozess lässt sich wie folgt veranschaufichen: Im Falle eines nicht-digitalen Beweismittels ist er sehr klar und intuitiv verständlich. Digitale Beweismittel sind jedoch von Natur aus für das menschliche Auge als solche nicht sichtbar. Daher muss das Beweismittel durch andere Hilfsmittel als das menschliche Auge verarbeitet werden. Da jeder Schritt eines solchen Prozesses den Einsatz eines bestimmten Werkzeugs oder Expertenwissen erfordert, muss der Prozess zuverlässig und reproduzierbar dokumentiert werden. Der Prozess selbst muss für den Richter und die Anwälte verständlich sein.

Die Sicherstellung von Beweismitteln ist sowohl ein juristisches als auch ein technisches Problem. In der Tat gibt es einen unmittelbaren Zusammenhang zwischen diesen beiden Aspekten. Das Gesetz spezifiziert, was unter welchen Umständen durch wen und von wo aus sichergestellt werden kann.

Tatsächlich stellt die Identifikation eines digitalen Beweismittels einen eigenen Prozess in drei Schritten dar:

1. Das Beweismittel muss in seiner physischen Form definiert werden können, so dass es auf einem konkreten Speichermedium gesichert ist.
2. Als Nächstes muss seine logische Position identifizierbar sein (wo es sich im Dateisystem befindet).
3. Zuletzt muss das Beweismittel in den richtigen Kontext gebracht werden, um seine Bedeutung erfassen zu können.

Jeder dieser Schritte erfordert technische Fähigkeiten und kann weiterhin die Bezeugung vor Gericht erfordern. Nachdem das physikalische Speichermedium in lesbare Daten übersetzt wurde, kann die Auswertung erfolgen. Diese beinhaltet sowohl eine technische als auch juristische Beurteilung. Daten, die in

ihren ursprünglichen Kontext gebracht wurden, werden Information genannt. Technische Schlussfolgerungen betreffen die Entstehung der Daten. (Wie, wann und durch wen wurden sie angelegt?) Die juristischen Beurteilungen betreffen die Relevanz der Information, ihre Zuverlässigkeit und wer sie bezeugen kann.

Carrier und Spafford 2003

Carrier u. Spafford (2003) vereinen den digitalen investigativen Prozess mit dem nichtdigitalen. Das Resultat ist der Integrated Investigation Process, der 17 Schritte beschreibt, welche in fünf Phasen zusammengefasst werden. Charakteristisch für dieses Modell ist, dass der physikalische Tatort parallel zum digitalen Tatort untersucht wird und die Ergebnisse der digitalen Ermittlung in die nicht-digitale Ermittlung einfließen. Im Folgenden fassen wir die fünf Phasen kurz zusammen.

1. Readiness Phases
   Ziel dieser Phase ist es sicherzustellen, dass die Einrichtung samt Infrastruktur jederzeit in der Lage ist, eine vollständige Untersuchung zu ermöglichen. Sowohl digitale als auch physische Beweismittel können verloren gehen, wenn sie nicht ordnungsgemäß sichergestellt und verwaltet werden. Diese Phase ist nicht an einen konkreten Vorfall gebunden und wird beständig durchgeführt.
2. Deployment Phases
   Diese Phasen sollen einen Mechanismus liefern, um Vorfälle erkennen und bestätigen zu können.
3. Physical Crime Scene Investigation Phases
   Aufgabe der Untersuchung eines physikalischen Tatortes ist es, Beweismittel zu sammeln und zu analysieren. Weiterhin muss versucht werden, den Tathergang zu rekonstruieren. Ziel einer digitalen Ermittlung ist es, Personen zu identifizieren, die für den Vorfall verantwortlich sind. Um diese Verbindung zwischen digitalen Beweismitteln und Personen herzustellen, sind physikalische Beweismittel erforderlich. Die einzelnen Schritte dieser Phase gleichen denen einer gewöhnlichen nicht-digitalen Ermittlung.
4. Digital Crime Scene Investigation Phases

Diese Phase beginnt, wenn digitale Geräte als physische Beweismittel sichergestellt wurden oder auf sonstige Art und Weise digitale Daten zur Analyse sichergestellt wurden. Während dieser Phase wird der Computer als zusätzlicher Tatort aufgefasst und ebenfalls nach Beweismitteln durchsucht. Alle Ergebnisse fließen wiederum in die Physical Crime Scene Investigation Phase ein.

5. Review Phase
   Die letzte Phase beinhaltet die rückblickende Überprüfung der gesamten Ermittlung, um Potential für Verbesserungen zu identifizieren. Das Ergebnis dieser Phase können neue Prozeduren, Fortbildungen oder Anschaffungen sein.

Baryamureeba und Tushabe 2004

Baryamureeba u. Tushabe (2004) schlagen eine Modifikation des von Carrier u. Spafford (2003) eingeführten Modells vor und beschreiben zwei zusätzliche Phasen: die Traceback- und Dynamite-Phasen, die versuchen, eine klare Trennung zwischen primärem Tatort (Computer) und sekundärem (physikalischem) Tatort zu schaffen. Das hiermit angestrebte Ziel ist es, die beiden Tatorte gleichzeitig analysieren zu können, um so Inkonsistenzen zu vermeiden.

Beebe and Clark 2005

Beebe u. Clark (2005) betrachten in ihrer Arbeit bereits vorgestellte Modelle und kommen zu dem Schluss, dass – während die meisten dieser Modelle sequentielle Modelle sind – die Prozesse eigentlich aber durchaus parallele Aufgaben beinhalten. Daher führen sie ein Modell ein, in dem sie unter anderem mehrere Unteraufgaben der Data Analysis Phase vorstellen. Diese Subphasen folgen dem SEE-Prinzip (Survey, Extract, Examine). Die Hauptphasen sind jedoch die aus den angeführten Modellen bereits bekannten. Weiterhin führen sie das Konzept der auf Ermittlungsziele ausgerichteten Analyseaufgaben ein. Beebe und Clark vergleichen ihr Modell sehr detailliert mit anderen bekannten Modellen in einer tabellarischen Übersicht.

Gong et al. 2005

Die Arbeit von Gong u. a. (2005) diskutiert die vorangegangenen Veröffentlichungen (Carrier u. Spafford, 2003; Beebe u. Clark, 2005) und führt einige neue Konzepte ein, beispielsweise den Begriff des Seek Knowledge (Suchintelligenz) als ein die gesamte Analyse bestimmender Faktor. Weiterhin identifizieren die Autoren einen Bedarf an Wiederverwendbarkeit des gewonnenen Wissens sowohl innerhalb eines Falles als auch vorfallübergreifend.

Ein anderes in dieser Arbeit eingeführtes Konzept ist der Begriff der Case Relevance, welche als diejenige Eigenschaft einer jeden Information definiert wird, die bemisst, inwieweit die Information dazu geeignet ist, die Fragen nach dem Wer, Was, Wo, Wann, Warum und Wie einer Ermittlung zu beantworten.

Kent et al. 2006

Kent u. a. (2006) beschreiben einen einfachen forensischen Prozess, der unabhängig von der konkreten Situation die folgenden Phasen umfasst:

- Collection
  In dieser Phase werden zunächst Daten aus allen potentiellen Quellen relevanter Daten identifiziert, beschriftet, erfasst und sichergestellt. Hierbei müssen Richtlinien zur Sicherung der Integrität der Daten befolgt werden. Diese Phase wird üblicherweise möglichst früh durchgeführt, um die Gefahr des Datenverlustes bei flüchtigen Daten zu minimieren.
- Examination
  Diese Phase umfassst die forensische Verarbeitung großer Mengen gesammelter Daten. Hier wird versucht, sowohl von Hand als auch automatisiert die relevanten Daten zu identifizieren und zu extrahieren. Die Integrität der Daten muss hierbei gewährleistet werden.
- Analysis
  Die nächste Phase des Prozesses ist es, die Ergebnisse der vorherigen Phase durch juristisch vertretbare Methoden und Techniken zu analysieren, um hinsichtlich der konkreten Fragestellung nützliche Informationen zu erhalten.

- Reporting
  Die letzte Phase besteht aus der Berichterstattung über die Analyseergebnisse. Der Bericht kann die verwendeten Methoden, eine Erläuterung über die Auswahl derselben sowie Empfehlungen für Verbesserungen der Richtlinien, Prozesse, Werkzeuge oder sonstiger Aspekte des forensischen Prozesses enthalten. Die Form und der konkrete Inhalt eines forensischen Berichtes richten sich stark nach der Art der Ermittlung (also etwa Strafverfolgung gegenüber unabhängiger Ermittlung).

Kiltz et al. 2009

Kiltz u. a. (2009) strukturieren ihr Modell entlang dreier „Achsen“:

1. Die zeitliche Abfolge der einzelnen Untersuchungsschritte.
2. Die Abstraktionsschicht, auf der die Untersuchung stattfindet, also beispielsweise Betriebssystemebene, Dateisystemebene, Anwendungsebene, etc.
3. Die Art der untersuchten Daten, beispielsweise Hauptspeicherdaten, Metadaten, Konfigurationsdaten, etc.

Die erste Achse entspricht den traditionellen Vorgehensweisen. Die anderen beiden Achsen machen das Vorgehen direkt abhängig von den Daten. Kiltz u. a. (2009) sprechen deshalb auch von einem ***datenzentrischen*** Vorgehensmodell.

5.6 Zusammenfassung

Vorgehensmodelle bilden einen abgesicherten methodischen Rahmen für Untersuchungen in der forensischen Informatik. In diesem Kapitel haben wir verschiedene Vorgehensmodelle vorgestellt. Näher betrachtet wurden etwa das Incident-Response-Vorgehensmodell, der investigative Prozess und das Common Model. Außerdem haben wir einen Überblick über weitere Modelle gegeben. Die Diskussion zeigt, dass es kein „bestes“ Vorgehensmodell gibt, sondern dass es vielmehr darauf ankommt, überhaupt ein Vorgehensmodell zugrunde zu legen.

Kapitel 6

Die Methodik der forensischen Informatik am Beispiel Partitionssysteme

Autoren: Michael Gruhn, Felix Freiling

In der forensischen Informatik treffen technische Fragestellungen auf rechtliche Fragestellungen, beispielsweise:

- Wie greife ich auf digitale Spuren zu, um ihren Beweiswert nicht zu schmälern?
- Was bedeuten digitale Spuren? Wie kann ich sicherstellen, dass die Spuren das bedeuten, was ich glaube?

Um diese Fragen zu beantworten, ist es notwendig, ein sehr gutes Verständnis für Recht und Technik zu besitzen, aber auch das Zusammenspiel dieser Bereiche muss in die Betrachtungen einbezogen werden. Wie sich diese Fragestellungen in der Praxis manifestieren, kann man am besten anhand von Beispielen erläutern. Dieser Studienbrief illustriert grundsätzliche methodische Fragen der forensischen Informatik anhand eines gut erforschten Bereichs der Datenträgerforensik.

Der größte Teil an digitalen Spuren fällt heute noch immer auf Datenträgern wie Festplatten oder USB-Sticks an. Bevor diese Spuren ausgewertet werden können, müssen zunächst die Datenträger selbst forensisch gesichert und ausgewertet werden. Hierzu muss der Forensiker den Aufbau und die Funktionsweise der entsprechenden Festplattentechnologie kennen. Außerdem muss er die erste Abstraktionsschicht auf Datenträgern, die Partitionssysteme, beherrschen. Beides ist Gegenstand dieses Kapitels.

Partitionssysteme sind ein wichtiges Konzept für die Organisation von Informationen auf Datenträgern. Mit Hilfe von Partitionen kann man einen Datenträger in mehrere eigenständige logische Bereiche aufteilen, die

unabhängig voneinander Daten verwalten können. Dies hat Vorteile etwa bei der Strukturierung von Daten (in System- und Benutzerdaten beispielsweise) oder bei der Nutzung eines Datenträgers mit mehreren Betriebssystemen (***dual boot***). Auch werden sogenannte swap-Partitionen verwendet, um Speicher für die Seitenauslagerung bereitzuhalten.

Partitionen sind omnipräsent und das Verständnis für Partitionssysteme ist eine wesentliche Voraussetzung für die Analyse von Datenträgern. Moderne Partitionssysteme sind aber einfach genug, um sie als einführendes Beispiel für methodische Fragen der forensischen Informatik nutzen zu können. Die Analyse der ***in*** den Partitionen gespeicherten Daten und ihrer Organisation (Dateisysteme) ist Teil der forensischen Untersuchung von Datenträgern. Dieses Thema werden wir hier nicht vertiefen. Interessierte verweisen wir auf Carrier (2005).

Ausblick

In diesem Kapitel betrachten wir zunächst allgemein Datenträgertechnologien und anschließend allgemeine Aspekte von Partitionen und Partitionssystemen. Anschließend werden wir zwei Partitionssysteme, nämlich DOS/MBR und GPT näher betrachten. Abschließend setzen wir uns noch mit grundsätzlichen Fragen des forensischen Zugriffs auf Datenträger auseinander.

6.1 Datenträgertechnologien

Es gibt viele unterschiedliche Technologien, auf denen Daten gespeichert werden können. Die wichtigsten und verbreitetsten werden im Folgenden vorgestellt.

6.1.1 Festplatten

Diese klassischen Festplatten bestehen im Inneren aus mehreren metallischen Scheiben, den sogenannten ***Platten*** (***platters***). Diese Platten sind mit einer magnetischen Eisenoxid- oder Kobaltschicht überzogen. Durch die Magnetisierung dieser Schicht werden die Daten auf den Platten gespeichert. Sowohl für das Schreiben (d.h. Magnetisieren) als auch das Lesen sind pro Platte

zwei sogenannte Schreib-/Leseköpfe zuständig. Diese Köpfe sind im Prinzip kleine Elektromagnete. Low Level Format erzeugt auf der Magnetoberfläche Strukturen für ***Spuren*** (***tracks***) und ***Sektoren*** (***sectors***) auf jeder Platte. Spuren werden von außen nach innen beginnend mit 0 durchnummeriert. Sektoren werden beginnend bei Nummer 1 durchnummeriert. Alle Spuren mit gleicher Nummer auf unterschiedlichen Scheiben bilden einen ***Zylinder*** (***cylinder***). Der Sektor ist die kleinste adressierbare Einheit der Festplatte.

6.1.2 Flash-Speicher

Im Gegensatz zu klassischen Festplatten mit beweglichen Platten verwenden moderne Speichermedien, wie z.B. USB-Sticks oder auch SSDs (***solid state drives***), Halbleiterbausteine um persistenten Speicher zu realisieren. Durch Hard- und Softwarestandards lassen sich solche Datenträger aber über dieselben Schnittstellen wie klassische Festplatten ansprechen.

Aus Sicht der forensischen Informatik ist es also bei einem Zugriff über diese Schnittstellen unerheblich, ob der Datenträger eine klassische (rotierende) Festplatte ist oder eine SSD. Trotzdem enthalten beide Klassen von Speichertechnologien auf Hardware-Ebene weitere und unterschiedliche Quellen von Spuren, die man sichern und analysieren kann.

Ein weiterer wichtiger Aspekt von SSDs ist das sogenannte ***wear leveling*** (dt. Abnutzungsausgleich). Da die Halbleiterbausteine der SSDs nur eine endliche Anzahl, 100.000 bis 5 Millionen, Schreibzyklen durchlaufen können bevor sie verschlissen sind, verwendet der Speichercontroller der SSD für Schreiboperationen immer eine andere Zelle. Somit werden die Schreiboperationen gleichmässig über alle Zellen verteilt. Der Speichercontroller verwaltet alle Zellen in etwa wie folgt: Die Zellen werden in Speicherblöcken gegliedert. Wird nun ein Speicherblock über- bzw. beschrieben, beschreibt der Controller nicht die Zellen genau dieses Speicherblocks, sondern sucht einen neuen unbenutzten Speicherblock mit wenig Schreibzyklen, beschreibt diesen und trägt diesen anstelle des alten Speicherblocks in seine Verwaltungsliste ein. Diesen Abstraktionsschritt nennt man das Flash Translation Layer. Dadurch werden alle Zellen gleichmäßig abgenutzt.

Der Fakt, dass die Dateninhalte der alten Speicherblöcke immer noch auf der SSD vorliegen, ist für die Forensik von großem Interesse (Regan, 2009). So ist sicheres Löschen durch Überschreiben, wie es bei normalen Festplatten verwendet wurde, bei SSDs ineffektiv (Wei u.a., 2011). Die Analyse wird aber

häufig von den proprietären Hardwarestandards erschwert (Billard u. Hauri, 2010). Der Zugriff auf diese Spuren kann jedoch nur unter Umgehung des Controllers der SSD erfolgen, wie zum Beispiel durch Auslöten der Speicherbausteine, da die Effekte des wear leveling ansonsten durch den Controller ja gerade transparent verborgen werden. Des Weiteren werden auf SSDs oft, ebenfalls zum wear leveling, spezielle Dateisysteme verwendet, die forensisch oft noch wenig analysiert sind (Zimmermann u.a., 2012).

6.1.3 Schnittstellen

Festplatten haben Schnittstellen, um den Zugriff auf die in ihnen gespeicherten Daten zu ermöglichen. Die Ein- und Ausgabe der Festplatte wird durch einen internen Controller geregelt. Über die Schnittstelle und ein passendes Kabel wird die Verbindung zum Computer hergestellt, wo ebenfalls ein Controller die Kommunikation regelt. Der Controller kann als Einsteckkarte realisiert sein oder sich bereits direkt auf dem Mainboard befinden und wird über eine entsprechende Logik in den physischen Speicherbereich des Rechners abgebildet. Die Kommunikation zwischen CPU und Festplatte erfolgt dann durch das Lesen und Schreiben dieser Speicherbereiche (speicherbasierte Ein-/Ausgabe).

Abbildung 6.1: Bild eines IDE-Anschlusskabels (Wikipedia, 2010).

ATA (AT Attachment) ist heute die gebräuchlichste Festplattenschnittstelle. Ursprünglich war die ATA-Schnittstelle nur zum Ansprechen von Festplatten

gedacht. Die Erweiterung zum AT Attachment Packet Interface (ATAPI) Standard ermöglichte später die Möglichkeit mit ATA-Befehlen auch andere Datenträger anzusprechen, solange sie diesen Standard unterstützen. Die Schnittstelle ist offiziell im ATA/ATAPI Standard (American National Standards Institute, 2002) festgelegt. ATA benötigt einen Controller, dieser ist aber meist bereits auf den Motherboards integriert.

Der ATA Standard legt die IDE (P-ATA) and S-ATA-Schnittstellen fest, welche im Folgenden genauer beschrieben werden.

ATA-3 erlaubte das Setzen von Passwörtern für die Ein-/Ausgabe. Erst nach Eingabe des Passwortes in spezielle Register können bestimmte ATA-Befehle ausgeführt werden. Mehr zu ATA-Passwörtern findet sich auch bei Carrier (2005, S. 36ff).

IDE (P-ATA)

Die Integrated Drive Electronics (IDE)-Schnittstelle, die ursprünglich von Western Digital entwickelt wurde, diente als Grundlage für den ATA/ATAPI Standard. „Integrated Drive Electronics“ bedeutet hier, dass die Elektronik zur Steuerung der Festplatte bereits in der Festplatte integriert ist.

Auf der Anschlussebene hat die IDE Schnittstelle 40 Pins (siehe Abbildung 6.1). Ein Controller befehligt bis zu zwei Platten an einem Kabel. Mit Einführung der S-ATA-Schnittstelle (Serial ATA) 2003 wurde IDE umbenannt zu P-ATA (Parallel ATA).

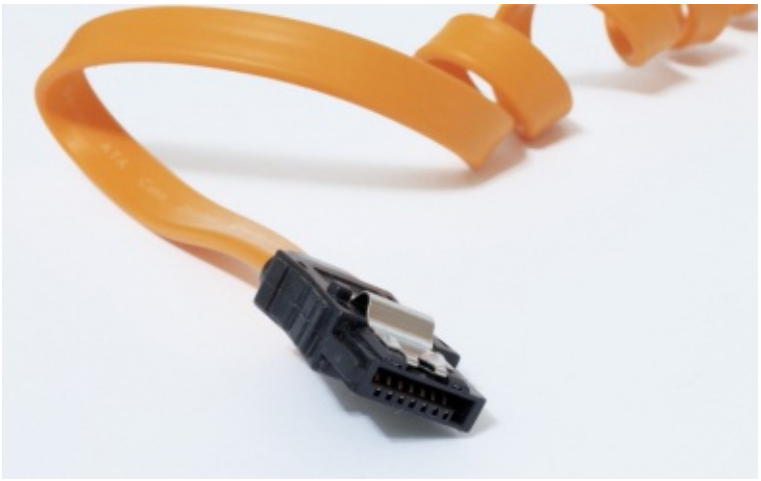


Abbildung 6.2: Bild eines S-ATA Datenkabels (Barroso, 2015).

Heute lassen sich mit Adaptern IDE-Festplatten an nahezu jede Schnittstelle anschließen (Netzwerk, USB etc.).

S-ATA (Serial ATA)

S-ATA (Serial ATA) stellt die Weiterentwickelung von P-ATA dar. S-ATA verfügt über schnellere Übertragungsraten und dedizierte Verbindungskabel pro Festplatte. Im Gegensatz zu P-ATA ist die Datenleitung vom Controller zum Datenträger nicht mehr parallel sondern seriell. Des Weiteren ist das Datenkabel getrennt vom Stromkabel (siehe Abbildung 6.2).

6.1.4 Host Protected Area (HPA)

Eine Festplatte stellt im Wesentlichen einen großen homogenen Speicherbereich für persistente Daten zur Verfügung, der durchgehend adressiert werden kann (LBA, siehe später). Aus praktischen Gründen wurden jedoch Bereiche definiert, die sich „am Ende“ der Festplatte befinden und die besondere Aufgaben haben.

Die Host Protected Area (HPA) ist ein spezieller Bereich von Datenträgern, den man leicht übersieht, denn dieser ist im normalen Betrieb nicht sichtbar. Die HPA wurde konzipiert zum Speichern von Konfigurationsdaten und anderen Daten, die nicht im normalen Betrieb zugreifbar seien sollen (American National Standards Institute, 1998, 6.12 Host Protected Area feature set). Viele Computerhersteller verwenden die HPA zum Hinterlegen von Daten, die zum Wiederherstellen des vorinstallierten Betriebssystems benötigt werden.

Der Speicherbereich der HPA liegt direkt nach dem normal sichtbaren Speicherbereich auf dem Datenträger. Die HPA, wenn vorhanden, kann mindestens mit den folgenden ATA-Befehlen gesteuert werden (American National Standards Institute, 2002, 6.15 Host Protected Area feature set):

- READ_NATIVE_MAX_ADDRESS_(EXT): Liest die maximale Größe des Datenträgers inklusive der HPA.
- SET_MAX_ADDRESS_(EXT): Setzt die maximale Größe des normal sichtbaren Datenträgers.

Mittels spezieller forensischer Werkzeuge wie dem Sleuthkit, EnCase oder FTK kann auf das Vorhandensein einer HPA geprüft werden.

Beispiel 29 (Linux-Befehle für HPA) ***Unter Linux kann der Datenträger*** `/dev/sdx` ***mittels des*** `hdparm` ***Befehls***

```
# hdparm -N /dev/sdx
```

auf das Vorhandensein einer HPA geprüft werden. Bei folgender Ausgabe

```
/dev/sdx:
max sectors = 4208152211/4271154564, HPA is enabled
```

ist die HPA aktiviert. Bei nicht aktivierter HPA lautet die Ausgabe:

```
/dev/sdx:
max sectors = 1342110815/1342110815, HPA is disabled
```

Mittels des Befehls

```
# hdparm -N x /dev/sdx
```

wird, bis zur nächsten Stromtrennung des Datenträgers, die maximale Größe des normal sichtbaren Bereichs auf `x` ***gesetzt***.

6.1.5 Device Configuration Overlay (DCO)

Seit ATA/ATAPI-6 gibt es die Device Configuration Overlay (DCO)-Funktionalität. Mit dieser kann die Konfiguration des Datenträgers nachträglich geändert werden. Von forensischem Interesse ist hierbei die Möglichkeit, die Größe des Datenträgers zu manipulieren. Dabei entsteht ein weiterer Speicherbereich hinter der HPA, der sogenannte Device Configuration Overlay (DCO)-Bereich. Diese Möglichkeit wird von Herstellern oft genutzt, um defekte Sektoren auszubessern, indem der Datenträger mehr Speicherkapazität hat, und dieser nach Bedarf in die DCO verschoben wird.

Die DCO kann mit folgenden ATA-Befehlen kontrolliert werden:

- DEVICE CONFIGURATION IDENTIFY: Fragt die derzeit verfügbaren Fähigkeiten des Datenträgers ab. Weicht diese Ausgabe von der Ausgabe des IDENTIFY DEVICE oder IDENTIFY PACKET DEVICE-Befehls ab, liegt ein DCO vor.

- DEVICE CONFIGURATION SET: Setzt oder entfernt Konfigurationen vom Datenträger. Mittels dieses Befehls kann auch die normal sichtbare Größe des Datenträgers geändert werden.
- DEVICE CONFIGURATION RESTORE: Entfernt das DCO permanent.

Wie bei der HPA kann mittels spezieller forensischer Werkzeuge wie dem The Sleuth Kit, EnCase oder FTK auf das Vorhandensein eines DCO geprüft werden.

Beispiel 30 (Linuxbefehle für DCO) ***Unter Linux kann der Datenträger*** `/dev/sdx` ***mittels des*** `hdparm` ***Befehls***

```
# hdparm --dco-identify /dev/sdx
```

auf das Vorhandensein eines DCO geprüft werden. Mittels des Befehls

```
# hdparm --dco-restore /dev/sdx
```

wird permanent das DCO entfernt. Dies kann zu Datenverlust führen.

6.1.6 Zugriff auf den Datenträger als Teil des Boot-Prozesses

Bei Computern mit IBM Personal Computer AT (PC-AT) kompatiblem BIOS, praktisch jedem Intel x86 Computer, lädt das BIOS den ersten 512 Byte Block des Boot Mediums an die Speicheradresse 0x7c00 (S.Tanenbaum, 1987) und springt dann mit der Ausführung an diese Adresse, so wie es in der BIOS Boot Specification (Compaq Computer Corporation, 1996, 6.5.1 Booting from BAIDs) definiert ist. Im BIOS des jeweiligen Computers kann das Boot Medium festgelegt werden bzw. die Reihenfolge, in der die Medien auf Bootfähigkeit geprüft werden sollen.

6.2 Partitionssysteme und ihre Analyse

6.2.1 Terminologie

Das Wort „Partition“ stammt aus dem Lateinischen und bedeutet „Teil“ oder „Stück“. Entsprechend versteht man unter dem Verb „partitionieren“ den Vorgang der Unterteilung, Trennung oder Zerstücklung. Dieser Begriff wird

verwendet, um die Aufteilung eines physischen Datenträgers in mehrere logische Teilbereiche zu bezeichnen. Diese logischen Teilbereiche werden dazu verwendet, um Daten mittels besonderer Verwaltungsstrukturen in Dateisystemen zu organisieren.

Wie oben bereits angemerkt sind Sektoren die kleinste adressierbare Einheit der Festplatte, d. h. der Speicherplatz einer Festplatte besteht aus einer Menge von Sektoren. Während der Begriff des Sektors noch stark mit der inneren Geometrie einer Festplatte verbunden ist (ein Sektor als ein Abschnitt einer Spur), verwendet man heute den etwas abstrakteren Begriff ***Block*** zur Bezeichnung der kleinsten adressierbaren Speichereinheit. Wir werden die Begriffe Sektor und Block synonym verwenden.

Auch für Festplatten und ihre Teilbereiche haben sich unterschiedliche Begriffe etabliert. Wir folgen hier der Terminologie von Carrier (2005) und werden für eine Festplatte den abstrakteren Begriff ***Laufwerk*** verwenden.

Definition 26 (Laufwerk und Partition) ***Ein*** Laufwerk ***(volume) ist eine Menge von adressierbaren Blöcken, die ein Betriebssystem oder eine Anwendung zur Speicherung von Daten verwenden kann. Eine*** Partition ***(partition) ist eine Menge von aufeinander folgenden Blöcken in einem Laufwerk***.

Gemäß der Definition ist eine Partition immer auch ein Laufwerk, aber ein Laufwerk ist nicht immer eine Partition. Laufwerke können mehrere Partitionen beinhalten.

6.2.2 Vor- und Nachteile von Partitionen

Die Gründe für die Partitionierung eines Datenträgers wurden bereits eingangs erläutert. Diese Gründe sind teilweise historischer Natur. Beispielsweise haben Dateisysteme wie etwa FAT eine Maximalgröße, die kleiner sein kann als die Gesamtgröße des Datenträgers. Mit Hilfe von Partitionierung lässt sich der Datenträger in logische Laufwerke aufteilen, welche jeweils die Maximalgröße des Dateisystems nicht überschreiten. Somit lässt sich der gesamte Speicherplatz des Datenträgers nutzen, indem die logischen Laufwerke das Datenträgers jeweils mit dem gegebenen Dateisystem formatiert werden.

Die logische Trennung in Partitionen erlaubt es, mehrere Betriebssysteme auf demselben Datenträger aber trotzdem unabhängig voneinander zu installieren

(***dual boot***), ohne dass sich die Betriebssysteme ein Dateisystem teilen und somit miteinander kooperieren müssen.

Während virtueller Speicher in Windows durch die Seitenauslagerung in eine Auslagerungsdatei realisiert ist, verwendet Linux hierfür eine eigene Partition, die sogenannte Swap-Partition. Der Vorteil dieses Ansatzes besteht darin, dass sich mehrere Betriebssysteme, die nie gleichzeitig laufen können, die Swap-Partition teilen können. Somit muss nicht jedes auf dem Datenträger installierte Betriebssystem Platz für seine eigenen Auslagerungsdateien im Dateisystem vorhalten.

Um einen Computer in den Ruhezustand (Hibernation) zu versetzen, also seinen Zustand einzufrieren, muss sein derzeitiger Zustand, also der gesamte Inhalt des RAMs, gespeichert werden. Hierfür kann ebenfalls eine zuvor angesprochene Swap-Partition verwendet werden.

Partitionen wurden in der Vergangenheit überdies hinaus auch verwendet, um die Auswirkungen von defekten Sektoren auf die Integrität von sich auf dem Datenträger befindenden Dateisystemen einzuschränken. So kann ein defekter Sektor in einer Partition das sich darin befindende Dateisystem zerstören, Dateisysteme in anderen Partitionen sind davon jedoch nicht betroffen. Da heutige Datenträger wesentlich weniger fehleranfällig sind, wird Partitionierung meist nicht mehr nur aus diesem Grund verwendet.

Des Weiteren verwenden viele Unix-Systeme eine Partition für Systemdateien und eine Partition für Benutzerdateien. Dadurch kann eine komplette System-Neuinstallation auf der Systempartition durchgeführt werden, ohne dass dabei die Benutzerdateien auf der Nutzerpartition verloren gehen.

Da die Metadaten einer Partitionierung, z. B. wo beginnt eine Partition und wo endet sie etc., ebenfalls auf dem Datenträger gespeichert werden müssen, kann dieser Speicherplatz nicht mehr für Anwenderdaten genutzt werden. Der durch Partitionierung verursachte Verschnitt ist jedoch im Allgemeinen so gering, dass dieser vernachlässigt werden kann.

Ein anderer Nachteil ist etwaiger Leistungsverlust beim Kopieren von einer Datei auf einer Partition mit eigenem Dateisystem des Datenträgers auf eine andere Partition mit anderem Dateisystem desselben Datenträgers, welcher beim Kopieren einer Datei in derselben Partition und somit demselben Dateisystem nicht auftritt.

6.2.3 Adressierungsarten

Physische Adressierung

Es gibt historisch bedingt unterschiedliche Arten, die Blöcke auf Laufwerken zu adressieren. Anfangs verwendete man eine hardwarenahe ***physische Adressierung***, die direkt vom physischen Aufbau des Datenträgers abgeleitet wurde. Hierzu nutzte man die ***Festplattengeometrie*** aus, zusammen mit der Beobachtung, dass ein Block auf der Festplatte eindeutig bestimmt war durch drei Angaben:

1. Die Angabe des Zylinders, auf dem sich der Block befand.
2. Die Angabe des Schreib-/Lesekopfes, den man ansprechen musste (hierdurch wurde die jeweilige Magnetscheibe ausgewählt).
3. Die Angabe des Sektors in der so selektierten Spur.

Daraus entstand die ***CHS-Adressierung***, die nach diesen drei „Koordinaten“ auf der Festplatte benannt ist (Zylinder-Kopf-Sektor, ***cylinder-head-sector***) , siehe auch Abbildung 6.3. Hierbei gilt es zu beachten, dass die Zählung bei den Sektoren jeweils bei 1 und bei Zylinder und Kopf bei 0 beginnt (American National Standards Institute, 1994, Abschnitt 7.1.2).

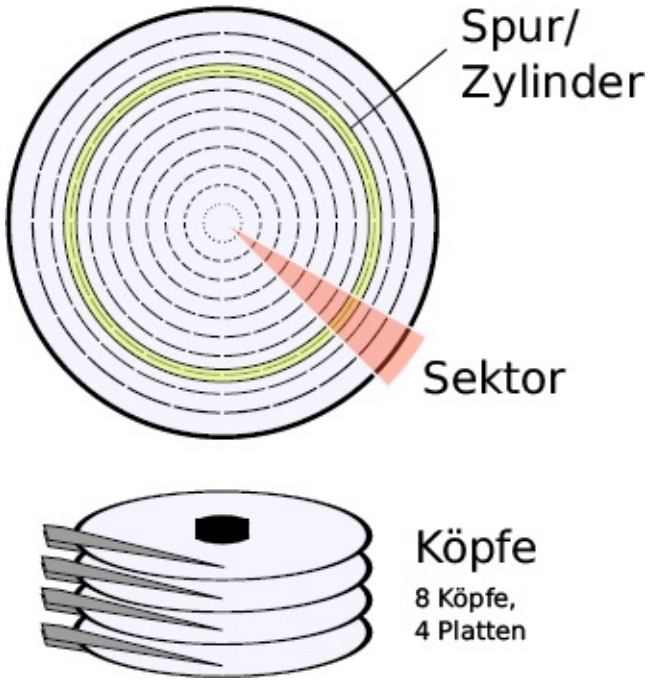


Abbildung 6.3: Illustration zur Zylinder-Kopf-Sektor (CHS)-Adressierung (Wikipedia, 2015).

Hersteller von Festplatten sind etwa um 1994 dazu übergegangen, die sogenannte ***Zone Bit Recording-Technik*** zu verwenden, bei der zum Rand der Platte aufgrund der längeren Spuren mehr Sektoren pro Spur verwendet werden. Dies führt dazu, dass bei modernen Festplatten die CHS-Adressierung vom BIOS bzw. Festplattencontroller in die tatsächliche Adresse auf der Platte übersetzt werden muss. Die CHS-Adressierung hatte also oft nichts mehr mit der physischen Adressierung zu tun (American National Standards Institute, 1994, 8.2 Translate mode). So wurde in 2002 mit ATA/ATAPI-6 die CHS-Adressierung gänzlich als obsolet deklariert (American National Standards Institute, 2002, 3.1.14 CHS (cylinder-head-sector)). Sie wird auch von modernen Datenträgern nicht mehr verwendet. Aus historischen Gründen sind CHS-Adressen aber in den meisten Datenstrukturen von Partitionen sowie in Dateisystemen immer noch vorhanden.

Logische Adressierung

An die Stelle der physischen Adressierung trat eine ***logische Adressierung***, die den physischen Aufbau des Datenträgers hinter einer Abstraktion verbirgt. In der logischen Adressierung werden alle Blöcke des Datenträgers unabhängig von ihrer physischen Lokalisierung mit einer fortlaufenden Nummer beginnend bei 0 durchnummeriert. Die Adresse eines Blocks nennt man dann die ***logische Blockadresse*** (***logical block address***, LBA).

Im Gegensatz zur physischen Adressierung hat die Adressierung mittels LBA den Vorteil, dass von der realen Geometrie des Speichermediums abstrahiert wird. Somit lässt sich mittels einer LBA ein Block in beliebigen Speichermedien adressieren, auch ohne Kenntnis über den internen Aufbau des Speichermediums.

Exkurs 12 (Umrechnung CHS in LBA) ***Auch wenn die logische Adresse konzeptionell vollkommen unabhängig ist von der physischen Adresse, so folgt die Vergabe von logischen Adressen aus technischen Gründen häufig einem bestimmten Muster: Auf einer Festplatte mit κ Köpfen pro Zylinder und σ Sektoren pro Spur kann die CHS-Adresse bestehend aus dem Tupel C (Zylinder), H (Kopf) und S_1 (Sektor) wie folgt in die LBA-Adresse x umgerechnet werden (American National Standards Institute, 2000, 6.2 Register delivered data transfer command sector addressing):***

$$x\ ((C \times \kappa) + H) \times \sigma + S - 1$$

Geometrisch kann man sich das wie folgt vorstellen:

- ***C Zylinder: Gibt die Kreisbahn, die sogenannte Spur auf der Platte an, auf der die Daten gespeichert sind***.
- ***S Sektoren: Gibt den Abschnitt auf der Spur der Platte an, in dem die Daten gespeichert sind***.
- ***H Kopf: Gibt die Platte an auf der die Daten gespeichert sind, d. h. den Lesekopf mit dem die Daten gelesen werden müssen***.

Das Problem bei dieser rein geometrischen Interpretation liegt klar auf der Hand. Die Kreisbahn, also die Spuren auf der Platte, sind zum Rand hin länger als in der Mitte. Bei einer einheitlichen Adressierung (also gleichbleibend vielen Sektoren pro Spur) würden zum Rand der Platte hin im Verhältnis weniger Daten pro Spur gespeichert werden. Dies liegt daran, dass der Umfang der Spur nach aussen hin, also mit größerem Radius, zunimmt. Des Weiteren wurden die ursprünglich gewählten Bitbreiten mit je 7 Bit für Kopf und Sektornummer zu klein bzw. falsch gewählt, da auf im Laufe der Zeit die Sektordichte auf den Platten immer größer wurde und die Kopfzahl auch nicht über zwei hinausging. Daher bezieht sich die CHS-Angabe oft bereits auf eine Adressübersetzungsabstraktion der Hardware anstatt auf die eigentliche geometrische Interpretation.

Basierend auf der LBA-Adresse kann man nun verschiedene logische Adressen eines Blocks definieren. Die englischen Bezeichnungen stammen von Carrier (2005, S. 74).

Definition 27 (Logische Laufwerks- und Partitionsadresse) ***Die*** logische Laufwerksadresse ***eines Blocks (logical volume address) ist die Adresse eines Blocks relativ zum Beginn des Laufwerks. Da jeder Datenträger ein separates Laufwerk ist, ist die logische Laufwerksadresse gleich der LBA-Adresse. Start- und Endblock einer Partition werden jeweils durch ihre logischen Laufwerksadressen spezifiziert***.

Die logische Partitionsadresse ***(logical partition volume address) ist die Adresse eines Blocks relativ zum Beginn der Partition. Falls ein Block in keiner Partition liegt, hat er keine logische Partitionsadresse***.

6.2.4 Partitionstabelle

Wie eingangs erwähnt, kann man mittels Partitionen ein Laufwerk in mehrere logisch separierte Abschnitte einteilen. Die Information über diese Einteilung wird in einer Datenstruktur gespeichert, die man in der Regel ***Partitionstabelle*** nennt. Die Partitionstabelle befindet sich üblicherweise „am Anfang“ des Datenträgers, üblicherweise im Block mit LBA-Adresse 0. Die Partitionstabelle enthält die wichtigsten Informationen zur Lokalisierung der Partitionen im Laufwerk, also Anfangs- und Endadressen bzw. Anfangsadresse und Länge der Partition. Normalerweise speichert die Partitionstabelle auch Informationen über die Art des Inhalts der Partition ab, z.B. ob es sich um eine bootbare Partition (mit Betriebssystem) handelt. Die Menge der in der Partitionstabelle abgelegten Daten wird durch das ***Partitionssystem*** definiert. Wir werden später die beiden wichtigsten Partitionssysteme vorstellen.

Analyse der Partitionstabelle

Bei der Laufwerksanalyse muss zunächst die Partitionstabelle geprüft werden. Wichtige Fragen dabei sind:

- Ist die Tabelle konsistent?
- Gibt es Blöcke, die keiner Partition angehören?

Die Konsistenzprüfung der Partitionstabelle sieht wie folgt aus. Partitionen in der Partitionstabelle müssen nicht entsprechend ihrem Layout auf der Platte sortiert sein. Daher sollten die Start- und Endsektoren auf Lücken überprüft werden da in den Lücken Daten versteckt sein können. In Lücken können sich auch Daten von vorhergehenden Dateisystemen befinden.

Beispiel 31 (Mögliche Partitionsanordnungen) ***Die Abbildung 6.4 zeigt eine Reihe von Beispielen, wie Partitionen prinzipiell über ein Laufwerk verteilt sein können. Durchgängig handelt es sich um ein Laufwerk mit zwei Partitionen (rot bzw. blau dargestellt), zwischen denen Lücken existieren können (grau). Die obere Reihe zeigt die „Standardfälle“, bei dem sich die beiden Partitionen nicht überlappen und allenfalls von einer kleineren Lücke voneinander absetzen. Die untere Reihe zeigt inkonsistente Fälle (von links nach rechts): Im ersten und zweiten Beispiel überlappen die beiden***

Partitionen, im zweiten Beispiel sogar derart, dass Partition 2 „in " Partition 1 liegt. Im letzten Beispiel läuft die Grenze von Partition 1 über das Ende des Laufwerks hinaus. Diese Fälle sind inkonsistent und weisen darauf hin, dass irgendetwas mit der Partitionstabelle nicht stimmt. Forensische Tools müssen diese Entartungen korrekt behandeln, denn auch in verschachtelten oder überlappenden Partitionen können Daten enthalten sein. Des Weiteren dürfen die Werkzeuge unter gar keinen Umständen abstürzen, z. B. wenn sie versuchen, über das physikalische Ende des Datenträgers hinaus zu lesen.

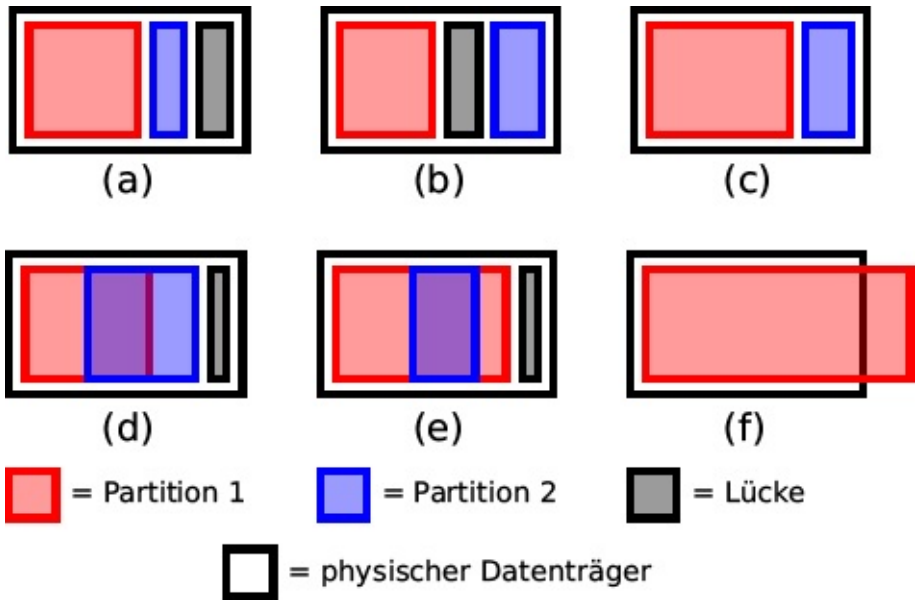


Abbildung 6.4: Mögliche Partitionsanordnungen.

Exkurs 13 (Größe einer Partition) ***Oft sind in der Partitionstabelle nur die Adressen des ersten und letzten Blocks abgespeichert. Bei der Berechnung der*** Größe einer Partition ***muss beachtet werden, dass sich sowohl der erste als auch der letzte Block noch in der Partition befinden. Das klingt trivial, führt aber immer wieder zu Problemen bei der Berechnung der Größe***.

Angenommen, wir haben die LBA-Adressen des ersten und letzten Blocks der Partition gegeben, also die Adresse S des Startblocks und die Adresse E des Endblocks einer Partition. Wir suchen die Größe G der Partition, also die Anzahl der Blöcke, die sich in der Partition befinden. Die naheliegende Berechnung von G durch die Differenz E – S ist aber falsch. Die richtige Berechnungsformel lautet dann:

$$G = E - S + 1$$

Man muss ja die letzte Zahl (den letzten Block der Partition) mitrechnen. Um dies zu sehen, betrachten sie das folgende hypothetische Berechnungsbeispiel:

Angenommen, der Startblock hat die Nummer S = 2 ***und der Endblock die Nummer E*** = 4. ***Die Partition besteht also aus den Blöcken mit den LBA-Adressen 2, 3 und 4, umfasst also drei Blöcke (G*** = 3***)***.

6.2.5 Analyse ohne Partitionsinformationen

Die Partitionstabelle enthält für die erste Analyse des Laufwerks sehr wichtige Informationen. Manchmal gehen diese Informationen durch Fehler verloren oder werden absichtlich gelöscht, um Spuren zu verwischen. In einem solchen Fall ist die Laufwerksanalyse deutlich erschwert, da die Standardtools keine Partitionen mehr anzeigen können. Man kann aber in vielen Fällen mit Heuristiken die Partitionen rekonstruieren.

Die typische Heuristik besteht darin, nach den typischen Mustern von Partitionsanfängen zu suchen. Da Partitionen selbst keine typischen Muster hinterlassen (außer in der Partitionstabelle), sind die gesuchten Muster genau genommen Hinterlassenschaften der in den Partitionen enthaltenen Dateisysteme. Grundlage für den heuristischen Ansatz ist also die Annahme, dass sich in jeder Partition, die aufgefunden werden soll, ein Dateisystem befunden hat. Der erste Block eines Dateisystems enthält oft einen typischen „Magic String“ oder andere Informationen, die den Anfang des Dateisystems identifizieren. Die folgenden Beispiele, die später noch ausführlicher erläutert werden, sollen dies illustrieren.

Beispiel 32 (Magic Strings) ***Der Master Boot Record (MBR) eines typischen für x86 formatierten Datenträgers enthält die Werte 0x55 und OxAA in den Bytes 510 und 511 des ersten Blocks***.

Der Boot Sektor des FAT32-Dateisystems enthält Felder für den OEM-Namen der Software, die das Dateisystem erstellt hat. Dieser Name ist bei Windows-Systemen häufig „MSWIN4.1“ und bei DOS-Systemen „MSDOS5.0“. Des Weiteren enthält FAT-32 einen Bezeichner für den Dateisystem-Typ am Anfang des Dateisystems. Typische Werte sind „FAT32“, „FAT16“ oder „FAT 12“.[6] Andere Dateisysteme enthalten ähnliche Magic Strings.

Wenn man weiß, an welcher Stelle innerhalb eines Blocks die typischen Muster der Magic Strings auftreten, kann man das Laufwerk systematisch danach absuchen und dadurch mögliche Kandidaten für Partitionen identifizieren. Diese müssen dann auf Plausibilität und Konsistenz geprüft werden (siehe oben).

6.2.6 Ausblick auf konkrete Partitionssysteme

Es gibt viele unterschiedliche Partitionssysteme. Wichtige Vertreter sind beispielsweise DOS/MBR, GPT, BSD Disklabel, Apple Partition Map (APM) und Volume Table Of Contents (VTOC). Im Folgenden werden nun zwei der am meisten verbreiteten Systeme im Detail vorgestellt. Die anderen Partitionssysteme werden in der Literatur (Carrier, 2005, Kapitel 6) vorgestellt. Mit den hier dargestellten Informationen sollte es aber auch möglich sein, neue und unbekannte Partitionssysteme selbst zu analysieren und zu dokumentieren.

Die Daten der Partitionssysteme werden ebenfalls dahingehend unterschieden, ob sie essentiell sind oder nicht. Dies geschieht zum einen aus einem theoretischen Blickwinkeln. Hierzu wird auf das Standardwerk von Carrier (2005) zurueckgegriffen. Zum anderen werden auch Ergebnisse aus eigener Analysen mit Linux (Ubuntu 13.10) und Windows 7 Betriebssystem bezüglich essentieller und nicht essentieller Daten gelistet.

6.3 DOS/MBR

Das zurzeit noch meist verbreitetste Partitionssystem ist der Master Boot Record (MBR), auch DOS-Partitionssystem genannt. Während es lange Zeit keine offizielle Spezifikation für das Layout des MBR gab und das Buch von Carrier (2005) oft als Defacto-Referenz zitiert wurde, sind mittlerweile Teile des MBR-Systems in der Unified Extensible Firmware Interface (UEFI)-Spezifikation (Unified EFI, Inc., 2013) dokumentiert, so dass dieser Abschnitt sowohl Carrier (2005) als auch die offizielle UEFI Spezifikation zum Erklären des MBR heranzieht.

MBR-Partitionen gehören zu den ältesten und kompliziertesten Partitionstypen. Carrier (2005) sagt hierzu:

> „They were originally designed in the 1980s for small systems and have been improved (i.e. hacked) to handle large modern systems.“

Diese von Carrier angesprochenen Erweiterungs-„Hacks“ sind wenig dokumentiert. Es gibt keine offizielle Spezifikation, auf die man sich berufen könnte. So ist das MBR-Format zwar in der UEFI-Spezifikation dokumentiert, jedoch ist diese Dokumentation unvollständig und geht beispielsweise nicht auf gebräuchliche Techniken wie erweiterte Partitionen ein. Da es also immer noch keinen vollständigen und offiziellen Standard des MBR-Systems gibt, wird in der Praxis in der Regel das Buch von Carrier (2005) als ***de facto***-Standard herangezogen.

Ein Datenträger, der mittels MBR partitioniert ist, beinhaltet in seinem ersten physischen Block den MBR. Der MBR ist 512 Byte groß. Die Felder des MBR können in Tabelle 6.1 auf Seite → eingesehen werden. Der MBR beinhaltet Partitionseinträge mit den Verweisen auf die einzelnen Partitionen. Abbildung 6.5 zeigt schemenhaft die eben erklärten Verweise. Im Folgenden werden die einzelnen Komponenten des MBR genauer beschrieben.

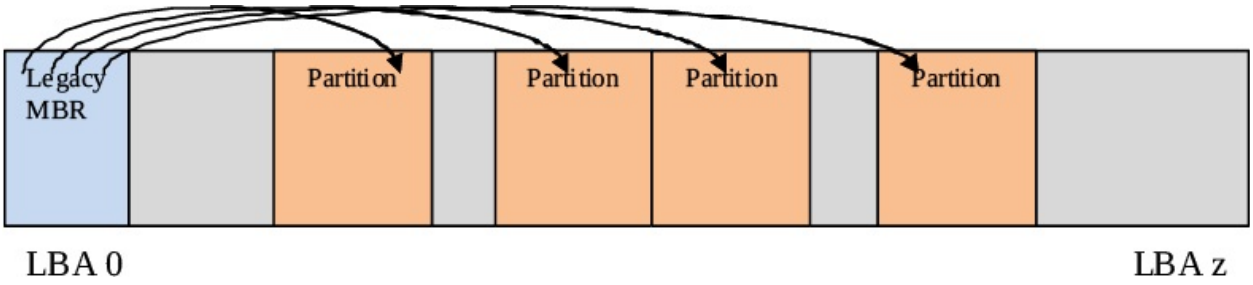


Abbildung 6.5: MBR (Unified EFI, Inc., 2013, Figure 16. MBRDisk Layout with legacy MBR example).

Die ersten 424 Byte des MBR (bzw. je nach Definition auch die ersten 440 oder sogar 446 Byte) sind reserviert für den ***boot code***. Der ***boot code*** ist der Teil des MBR, der vom BIOS in den Hauptspeicher geladen und dort ausgeführt wird. Da der ***boot code*** maximal 446 Byte groß sein kann, besteht die erste Aufgabe des ***boot codes*** in der Regel darin, anderen Programmcode nachzuladen (in diesem Fall spricht man von einem ***multi stage bootloader***) oder ***boot code*** aus den vom MBR referenzierten Partitionen zu laden und auszuführen (hier spricht man von einem ***chain loading bootloader***).

Exkurs 14 (Bootloader) ***Bootloader sind in der Regel als Teil des Betriebssystems realisiert. In diesem Fall liegt der Code des Bootloaders innerhalb einer Partition. Es ist aber auch möglich, dass Bootloader als Teil des boot codes im MBR realisiert sind. Hierbei werden meist zusätzlich einer***

oder mehrere Sektoren hinter dem MBR, aber vor ***dem Beginn der ersten Partition, für zusätzlichen Code genutzt***.

Der standardmäßige boot code von Microsoft analysiert die Partitionstabelle und identifiziert die erste bootbare Partition. Daraufhin lädt er den ersten Block dieser Partition und führt ihn aus.

Boot-Sektor-Viren nisten sich in die ersten 446 Bytes des MBR ein und werden dadurch bei jedem Systemstart ausgeführt.

Bytes 440 bis 443 des MBR werden von UEFI als ***eindeutige Datenträger-Signatur*** definiert. Diese Signatur kann vom Betriebssystem verwendet werden, um Datenträger voneinander zu unterscheiden. In Carrier (2005) wird diese Signatur jedoch nicht referenziert und sie wurde in den Anfängen des MBR auch nicht als eine solche verwendet; dieses Feld war seinerzeit Teil des Boot Codes, der dann 446 Byte umfasste.

Bytes	Beschreibung	Essentiell (Carrier, 2005)	Essentiell Linux	Essentiell Windows
0 - 424... 446	Boot Code	Nein	Nein[1]	Nein[1]
440 - 443	Eindeutige Datenträger-Signatur	?	Nein	Nein
446 - 461	Erster Eintrag der Partitionstabelle	Nein	Ja	Ja
462 - 477	Zweiter Eintrag der Partitionstabelle	Nein	Ja	Ja
478 - 493	Dritter Eintrag der Partitionstabelle	Nein	Ja	Ja
494 - 509	Vierter Eintrag der Partitionstabelle	Nein	Ja	Ja
510 - 511	Boot-Signatur	Nein	Ja	Ja[2]
512 - Logische Blockgröße	Reserviert	Nein	Nein	Nein

[1] Zum Booten.

[2] Betriebssystem will Datenträger reparieren.

Tabelle 6.1: MBR (Unified EFI, Inc., 2013, Table 13. Legacy MBR).

Danach folgen die vier ***Einträge der Partitionstabellen***, die jeweils 16 Byte umfassen. Diese geben jeweils unter anderem den Start und das Ende der jeweiligen Partition auf dem Datenträger an. Die Felder eines Partitionstabelleneintrages sind in Tabelle 6.2 aufgeführt. Ihre Bedeutung wird in der nächsten Sektion genauer erklärt.

An Byte Offset 510 befindet sich die ***Boot-Signatur***. Diese besteht aus den Bytes 0x55 an Offset 510 und OxAA an Offset 511.

Exkurs 15 (Verwendung der Boot-Signatur) ***Die Boot-Signatur wird vom PC-AT BIOS verwendet, um einen bootbaren Datenträger zu identifizieren. Soll ein Datenträger von einem PC-AT BIOS gebootet werden, dann überprüft das PC-AT BIOS zuerst diese Signatur. Ist diese Signatur vorhanden, dann lädt es die ersten 512 Byte, d. h. den gesamten MBR, vom Datenträger an die Speicheradresse 0x7C00 des PCs und springt mit der Ausführung blind an diese Adresse (S.Tanenbaum, 1987)***.

Dieser Exkurs macht ebenfalls deutlich, warum der MBR genau 512 Byte lang ist und sich an seinem Anfang ausführbarer Code befindet.

Die restlichen Bytes bis zur vollen logischen Blockgröße sind laut UEFI-Spezifikation reserviert und genullt.

MBR Partitionstabellen-Eintrag

Bytes	Beschreibung	Essentiell (Carrier, 2005)	Essentiell Linux	Essentiell Windows
0 - 0	Boot-Indikator	Nein	Nein[1]	Nein[1]
1 - 3	CHS Start-Adresse	Ja / Nein (wenn LBA)	Nein	Nein
4 - 4	Partitionstyp	Nein	Jein[2]	Jein[2]
5 - 7	CHS End-Adresse	Ja / Nein (wenn LBA)	Nein	Nein
8 - 11	LBA Start-Adresse	Ja	Ja	Ja
12 - 15	Größe in LBA-Blöcken	Ja	Ja	Ja

[1] Zum Booten.

[2] Zum auto-mounten.

Tabelle 6.2: MBR Partitionstabellen-Eintrag Unified EFI, Inc. (2013).

Der Wert 0x80 im ***Boot-Indikator***-Feld zeigt an, dass diese Partition gebootet werden kann. Jeder andere Wert bedeutet, dass die Partition nicht gebootet werden kann.

Die ***CHS Start-Adresse*** gibt den Start-Sektor der Partition in CHS-Form an. Die ***CHS End-Adresse*** gibt den letzten Sektor der Partition in CHS-Form an. Die UEFI-Spezifikation (Unified EFI, Inc., 2013) sagt, dass diese Felder nicht verwendet werden sollen sondern nur die Felder im LBA-Format gültig sind. Doch viele (besonders ältere) Programme und Betriebssysteme verwenden weiterhin die Felder im CHS-Format. Die CHS-Adresse ist in die 3 Byte der CHS-Adressfelder kodiert (wie in Abbild 6.6 dargestellt).

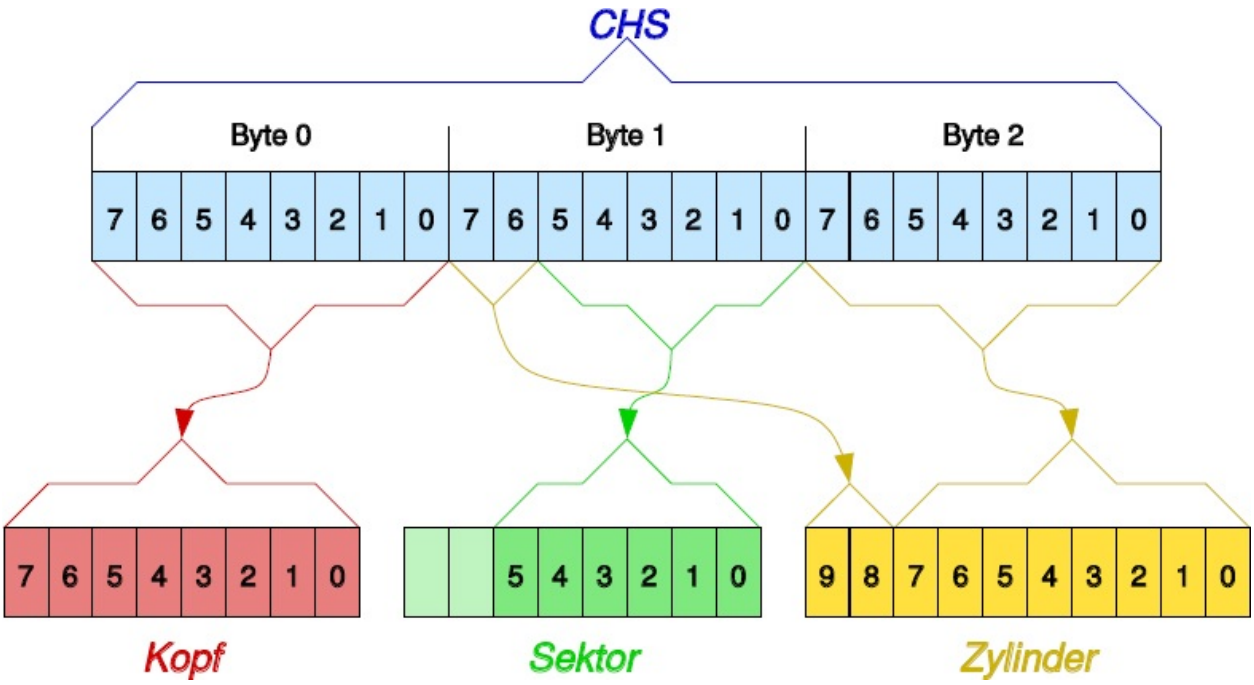


Abbildung 6.6: CHS-Adressierung wie in MBR kodiert (Wikipedia, 2008).

Der ***Partitionstyp*** gibt den Typ der Partition an. Dieser Typ ist meist gleichzusetzen mit dem Dateisystem, das sich in dieser Partition befindet, oder mit der Art der Verwendung der Partition, z. B. der Verwendung als Linux Swap-Partition. Tabelle 6.3 auf der nächsten Seite listet einige ausgewählte Beispiele von Partitionstypen auf.

Der Eintrag ***LBA Start Adresse*** enthält die Start-Adresse der Partition im LBA-Format. Die ***Größe in LBA-Blöcken*** gibt an, wie viele LBA-Blöcke die Partition lang ist. Alle Felder außer die CHS-Adressfelder verwenden Little-Endian-Byte-Reihenfolge (Unified EFI, Inc., 2013, 1.8.1 Data Structure Descriptions).

Beispiel 33 (Formatierung unter Linux) ***Der Prozess zum Erstellen, Partitionieren, Formatieren und Mounten eines Datenträgerabbildes für Linux sieht in etwa wie folgt aus:***

```
$ dd if=/dev/zero of=test.img bs=1M count=100
$ fdisk -b 512 -C 1048576 test.img
# kpartx -a -v test.img
# mkfs.vfat /dev/mapper/loop0p1
# mkfs.vfat /dev/mapper/loop0p2
# mount /dev/mapper/loop0p1 test
```

```
# umount /dev/mapper/loop0p1
```

ID	Name
0x01	FAT12
0x04	FAT16 ¡32M
0x05	Erweiterte Partition
0x06	FAT16
0x07	HPFS/NTFS
0x0b	W95 FAT32
0x0c	W95 FAT32 (LBA)
0x0e	W95 FAT16 (LBA)
0x0f	W95 Ext’d (LBA)
0x11	Hidden FAT12
0x14	Hidden FAT16 ¡32M
0x17	Hidden HPFS/NTF
0x1b	Hidden W95 FAT32
0x1c	Hidden W95 FAT32 (LBA)
0x82	Linux Swap
0x83	Linux
0x8e	Linux LVM
0xee	GPT
0xef	EFI (FAT-12/16/32)

Tabelle 6.3: Liste ausgewählter Partitionstypen (Unified EFI, Inc., 2013, 5.2.2 OS Types).

6.3.1 Primäre und Sekundäre Partitionen

Ursprünglich unterstützte der MBR nur maximal vier Partitionen. Doch dies ist manchmal zu wenig für moderne Systeme. Deshalb gibt es eine inoffizielle Erweiterung, sogenannte ***erweiterte Partitionen***. Die Idee der Erweiterung besteht daraus, im MBR ein, zwei bis maximal drei Partitionseinträge für „normale Partitionen“ zu erstellen und alle weiteren Partitionen in eine „erweiterte“ Partition zu packen.

Hierbei unterschiedet man folgende Begriffe:

- ***primäre Dateisystempartition*** (***primary file system partition***): Dies ist eine Partition im MBR mit Dateisystem.
- ***primäre erweiterte Partition*** (***primary extended partition***): Dies ist eine Partition im MBR, an deren Beginn eine weitere Partitionstabelle steht.
- ***sekundäre Dateisystempartition*** (***secondary file system partition***): Dies ist eine Partition innerhalb einer primären erweiterten Partition mit einem Dateisystem. Diese werden in Windows auch ***logische Partitionen*** genannt. Ihr Aufbau ist derselbe wie der von primären Dateisystempartitionen, mit dem Unterschied, dass sie innerhalb einer erweiterten Partition liegen.
- ***sekundäre erweiterte Partition*** (***secondary extended partition***): Dies ist eine Partition mit einer Partitionstabelle und einer sekundären Dateisystempartition. Diese bilden sozusagen einen „Wrapper“ um die sekundäre Dateisystempartitionen.

6.3.2 Sonderfälle

In der UEFI-Spezifikation ist zwar definiert, dass es keine Partition geben darf die eine andere Partition, wie in Bild 6.4 (d) und (e) dargestellt, überlappt („Each partition must not overlap with other partitions.“ (Unified EFI, Inc., 2013, 5.2.1 Legacy Master Boot Record (MBR))) oder deren Start- bzw. End-Adresse außerhalb des Datenträgers liegt („The partition defined by each MBR Partition Record must physically reside on the disk (i.e., not exceed the capacity of the disk).“ (Unified EFI, Inc., 2013, 5.2.1 Legacy Master Boot Record (MBR))). Da das MBR-Format jedoch lange Zeit gänzlich undokumentiert war, erkennen manche Betriebssysteme solche und auch anders entartet partitionierte Datenträger oftmals als valide. Dies kann zu Verwirrungen in forensischen Tools führen, da nicht klar definiert werden kann, ob und inwieweit die vorliegende Partitionierung valide ist und wie diese vom ursprünglichen Betriebssystem interpretiert wurde. Dies ist besonders bei sekundären Partitionen problematisch, da diese weder im UEFI noch in einem anderen Standard offiziell dokumentiert sind.

6.4 Globally Unique Identifier (GUID) Partition Table (GPT)

Die Globally Unique Identifier (GUID) Partition Table (GPT) ist Teil des Unified Extensible Firmware Interface (UEFI) Standards. UEFI definiert eine neue zentrale Schnittstelle zwischen der Computersoftware (Betriebssystem) und der Computerfirmware (Hardware) (Unified EFI, Inc., 2013, 1 Introduction). Das erklärte Ziel von UEFI ist es, das veraltete PC-AT BIOS zu ersetzen. Innerhalb dieses Vorhabens kann GPT als Nachfolger zum klassischen MBR-Verfahren angesehen werden. Wir betrachten nun GPT unter forensischen Gesichtspunkten und vertiefen dabei Einblicke aus bisherigen Arbeiten, die GPT untersucht haben (Nikkel, 2009; Carrier, 2005).

GPT erlaubt es, Festplatten und Partitionen größer als 2 TiB zu verwalten. Des Weiteren erlaubt GPT das Verwalten von bis zu 128 Partitionen. Es verwendet 64-Bit-LBA-Adressen und enthält redundante Kopien wichtiger Datenstrukturen zur Fehlertoleranz.

Ein GPT-partitionierter Datenträger besteht jeweils aus fünf Bereichen: Protective MBR (PMBR), GPT Header, GPT-Partitionstabelle, Partition Area und der Backup Area. Der PMBR wird hierbei zur Abwärtskompatibilität mit älteren Werkzeugen genutzt. Dieser PMBR verhindert das versehentliche Löschen einer GPT Partition durch solche ältere Werkzeuge. Der in LBA 0 liegende PMBR verweist auf den mit GPT-partitionierten Datenträgerbereich, der an LBA 1 beginnt. Der darin enthaltene GPT Header definiert die eigentlichen GPT Partitionen.

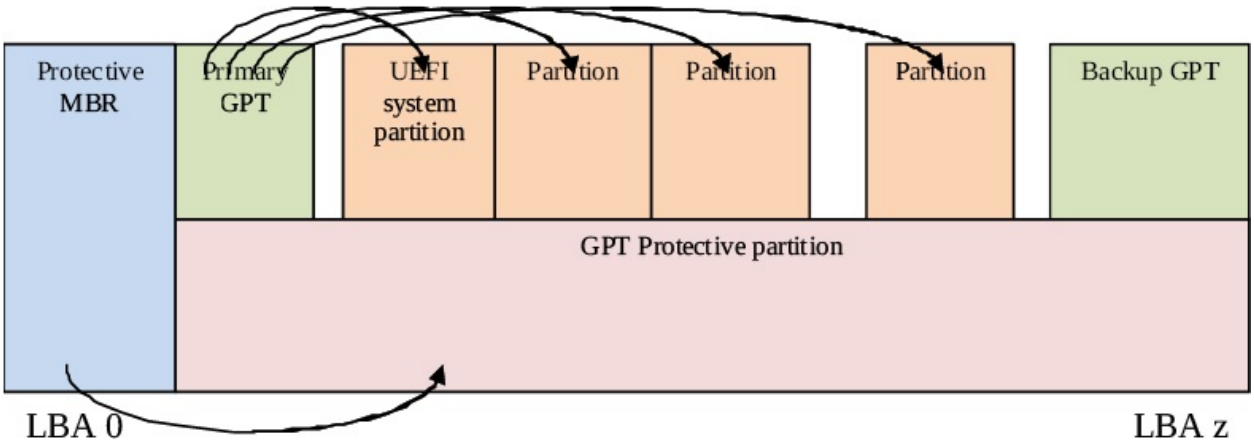


Abbildung 6.7: GPT Layout mit PMBR (Unified EFI, Inc., 2013, Figure 17. GPT disk layout with protective MBR example).

Abbildung 6.7 zeigt schemenhaft die eben erklärten Verweise. Im Folgenden werden diese Teile nun genauer beschrieben.

6.4.1 Protective MBR (PMBR)

Um Rückwärtskompatibilität zu gewährleisten, kann ein mit GPT partitionierter Datenträger an LBA 0 einen sogenannten Protective (dt. Schützenden) MBR (PMBR) enthalten. Dieser PMBR dient dazu, den Datenträger für ältere Software-Anwendungen als belegt zu markieren und somit vor versehentlicher Formatierung zu schützen. Der PMBR ist wie der MBR aufgebaut, hat aber nur eine Partition vom Typ EFI (0xee). Die restlichen Partitionseinträge sind ausgenullt. Eine komplette Darstellung des PMBR ist in Tabelle 6.4 auf der nächsten Seite zu sehen.

Der im PMBR verwendete PMBR Partitionstabelleneintrag ist in Tabelle 6.5 auf der nächsten Seite aufgeführt. Die essentiellen Daten sind analog zum klassischen MBR.

Der PMBR Partitionstabelleneintrag ist als nicht bootfähig deklariert. Des Weiteren verweist sowohl seine CHS als auch seine LBA Start-Adresse auf den zweiten logischen Block. Der Partitionstyp ist 0xEE, was GPT entspricht (vgl. Tabelle 6.3). Sowohl die CHS End-Adresse als auch die Größe in LBA-Blöcken muss so gewählt werden, dass die Partition sich bis einschließlich des letzten Blocks des Datenträgers erstreckt. Die Essentiellen Daten sind ebenfalls analog zum klassischen MBR.

Bytes	Beschreibung
0 - 439	Boot Code (von UEFI nicht verwendet)
440 - 443	Eindeutige Datenträger-Signatur (nicht verwendet; genullt)
444 - 445	Unknown (genullt)
446 - 461	GPT PMBR Partitionstabelleneintrag (siehe Tabelle 6.5)
462 - 477	Zweiter Eintrag der Partitionstabelle (ausgenullt)
478 - 493	Dritter Eintrag der Partitionstabelle (ausgenullt)
494 - 461	Vierter Eintrag der Partitionstabelle (ausgenullt)
510 - 511	Boot-Signatur
512 - Logische Blockgröße	Reserviert

Tabelle 6.4: Protective MBR (Unified EFI, Inc., 2013, Table 15. Protective MBR).

Bytes	Beschreibung
0 - 0	Boot Indikator (0x0)
1 - 3	CHS Start-Adresse (0x000200)
4 - 4	Partitionstyp (0xEE)
5 - 7	CHS End-Adresse (Letzter Block des Datenträgers oder 0xFFFFFF falls Größe des Datenträger Wortbreite des Feldes überschreitet)
8 - 11	LBA Start-Adresse (0x00000001)
12 - 15	Größe in LBA-Blöcken (”Größe des Datenträgers in LBA 1 oder 0xFFFFFFFF falls Größe des Datenträger Wortbreite des Feldes überschreitet)

Tabelle 6.5: PMBR Partitionstabelleneintrag (Unified EFI, Inc., 2013, Table 16. Protective MBR Partition Record protecting the entire disk).

6.4.2 GPT Header

Bytes	Beschreibung	Essentiell (Carrier, 2005)	Essentiell Linux	Essentiell Windows 7	Essentiellmmls
0 - 7	EFI-Signatur „EFI PART“	Nein	Ja[1]	Ja[1]	Ja[1]
8 - 11	Revision	Ja	Ja	Ja	Nein
12 - 15	GPT Header Größe in Bytes	Ja	Ja	Ja	Nein
16 - 19	Header CRC32-Prüfsumme	Nein	Ja	Ja	Ja
20 -	Reserviert	Nein	Nein	Nein	Nein

20 - 23	Reserviert (ausgenullt)	Nein	Nein	Nein	Nein
24 - 31	LBA des Headers	Nein	Nein	Nein	Nein
32 - 39	LBA des GPT Backup Header	Nein	Nein	Nein	Nein
40 - 47	LBA des ersten Blocks der Part. Area	Ja	Ja	Ja	Ja
48 - 55	LBA des letzen Blocks der Part. Area	Nein	Nein	Nein	Nein
56 - 71	Datenträger-GUID	Nein	Nein	Nein	Nein
72 - 79	LBA des ersten Blocks der Part.-Tabelle	Ja	Ja	Ja	Ja
80 - 83	Anzahl der Partitionstabellene inträge	Ja	Jein[2]	Jein[2]	Jein[2]
84 - 87	Größe eines Partitionstabellene intrages	Ja	Ja	Ja	Ja
88 - 91	CRC32-Prüfsumme der Partitionstabelle	Ja	Ja	Ja	Ja
92 - Blocksize	Reserviert (ausgenullt)	Nein	Nein	Nein	Nein

[1] Ohne wird der PMBR verwendet.

[2] Muss größer als die zu verwendende Partitionsnummer sein.

Tabelle 6.6: GPT Header (Unified EFI, Inc., 2013, Table 17. GPT Header).

Der GPT Header liegt an LBA 1 (Unified EFI, Inc., 2013, 5.3.1 GPT overview). Er definiert Größe und Lage der eigentlichen Partititionstabellen.

Seine genaue Struktur ist in Tabelle 6.6 dargestellt. Die einzelnen Felder des GPT Headers werden nun im einzelnen erklärt.

Der GPT Header fängt mit der ***EFI-Signatur*** an. Diese besteht aus 8 Bytes, die den ASCII-String „EFI PART“ enthalten. Das ***Revision***-Feld gibt an, welche GPT-Version vorliegt. Version 1.0 wird als 0x00010000 angegeben. Als Nächstes folgt die Angabe der ***GPT Header-Größe in Bytes***. Die ***Header CRC32-Prüfsumme*** ist eine CRC32-Prüfsumme über den gesamten GPT Header, dessen Größe durch ***GPT Header-Größe in Bytes*** gegeben ist. Für die Berechnung der Prüfsumme wird das Feld der ***Header CRC32-Prüfsumme*** auf 0 gesetzt. Alle als ***reserviert*** markierten Bereiche müssen laut UEFI-Spezifikation ausgenullt werden. Die meisten Programme schenken diesen Bereichen jedoch sowieso keinerlei Bedeutung. Die ***LBA des Headers*** selbst ist ebenfalls im Header gespeichert, so dass hier abermals eine Konsistenzprüfung stattfinden kann. GPT verwendet, wie bereits erwähnt, zusätzlich zum Header am Beginn einen weiteren Backup Header, der sich in der letzten LBA des Datenträgers befindet. Die ***LBA des GPT Backup Header*** in der Backup Area wird auch im GPT Header gespeichert. Die ***LBA des ersten Blocks der Partition Area*** und die ***LBA des letzten Blocks der Partition Area*** werden im Header gespeichert; dadurch kann die Partition Area gefunden werden. Bei GPT wird jeder Datenträger durch eine Datenträger-GUID identifiziert. Eine GUID ist ein globaler eindeutiger 128 Bit langer zufällig gewählter Identifikator der den jeweiligen Datenträger eindeutig identifizieren soll.

Des Weiteren enthält der Header Informationen über

- den LBA des ersten Blocks der Partitionstabelle,
- die Anzahl der Partitionstabelleneinträge und
- die Größe eines Partitionstabelleneintrages.

Windows beschränkt die Zahl der Einträge in der Partitionstabelle auf 128. Der Header enthält darüber hinaus noch eine weitere Checksumme zur Fehlererkennung und -korrektur. Dies ist die ***CRC32-Prüfsumme der Partitionstabelle***. Sie wird verwendet um die Korrektheit der Partitionstabellen sicherzustellen. Die CRC32-Prüfsumme wird berechnet vom ***LBA des ersten Blocks der Partition Area*** und erstreckt sich über die folgenden ***Anzahl der Partitionstabelleneinträge*** multiplizierten mit ***Größe eines Partitionstabelleneintrages*** Bytes. Das Format der Partitionstabelleneinträge ist

in Tabelle 6.7 auf Seite → dargestellt und wird im nächsten Abschnitt genauer beschrieben.

Der Header von UEFI-Version 1.0 ist insgesamt 92 Byte groß. Die restlichen nicht genutzten Bytes des logischen Blocks werden mit Nullen ausgefüllt. Wie der MBR verwendet auch GPT die Little-Endian-Byte-Reihenfolge (Unified EFI, Inc., 2013, 1.8.1 Data Structure Descriptions).

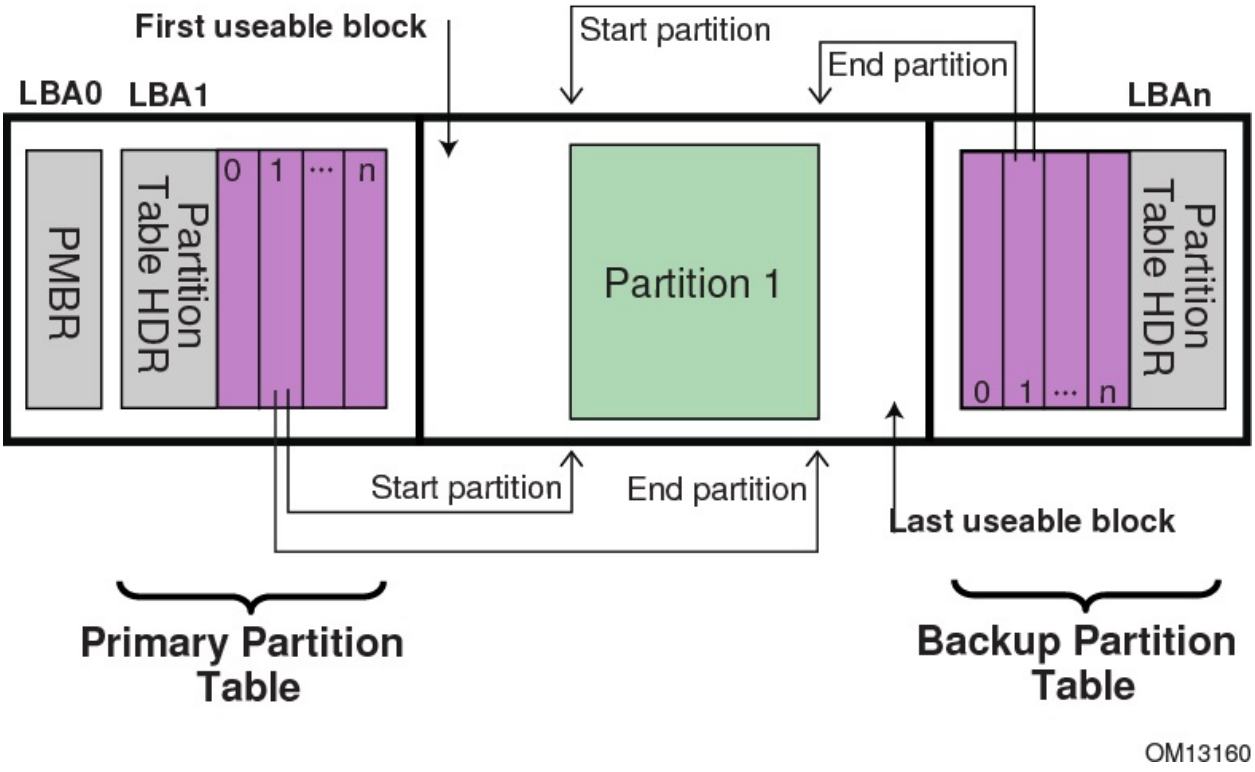


Abbildung 6.8: GPT Layout-Beispiel (Unified EFI, Inc., 2013, Figure 19. GUID Partition Table (GPT) example).

Laut UEFI-Spezifikation müssen zum Festellen der Korrektheit eines GPT Headers folgende Daten geprüft werden (Unified EFI, Inc., 2013, 5.3.2 GPT Header):

- ***EFI-Signatur*** muss „EFI PART“ enthalten
- ***Header CRC32-Prüfsumme*** muss valide sein
- ***LBA des Headers*** enthält wirklich den Header selbst
- ***CRC32-Prüfsumme der Partitionstabelle*** muss valide sein

Falls die GPT die Haupttabelle an LBA 1 ist, muss noch zusätzlich geprüft werden, ob der ***LBA des GPT Backup Header*** auch wirklich einen validen GPT Header enthält. Diese Prüfung erfolgt auch aufgrund der obigen Kriterien. Falls der GPT Header korrupt ist, muss der Backup Header verwendet werden und mit Hilfe dessen der Haupt-GPT Header an LBA 1 wieder rekonstruiert werden. Die exemplarischen Verknüpfungen inklusive Backup Header sind in Abbildung 6.8 illustriert.

Bytes	Beschreibung	Essentiell (Carrier, 2005)
0 - 15	Partitionstyp-GUID	Nein
16 - 31	Partitions-GUID	Nein
32 - 39	Start-LBA der Partition	Ja
40 - 47	End-LBA der Partition	Ja
48 - 55	Attribute (siehe Tabelle 6.9)	Ja
56 - 127	Partitionsname (Nullterminierter String)	Nein
128 - „Größe eines Partitionstabelleneintrages“	Reserviert (ausgenullt)	Nein

Tabelle 6.7: GPT Partitionseintrag (Unified EFI, Inc., 2013, Table 18. GPT Partition Entry).

6.4.3 GPT Partitionseintrag

Ein GPT-Partitionseintrag, wie in Tabelle 6.7 dargestellt, enthält am Anfang einen ***Partitionstyp-GUID***. Dieser identifiziert den Partitionstyp eindeutig. Mögliche GUIDs von Partitionstypen sind in Tabelle 6.8 auf der nächsten Seite gelistet. Der ***Partitionstyp-GUID*** ist in seiner Funktion vergleichbar dem Partitionstyp-Feld des MBR. Mit ihm lassen sich Partitionstypen identifizieren.

So muss sich beispielsweise jeder Betriebssystem-Hersteller seinen eigenen Partitionstyp-GUID definieren. Auf die Partitionstyp-GUID folgt die ***Partitions-GUID***. Der dort abgelegte Wert soll die ***Partition*** eindeutig identifizieren.

Als Nächstes folgt die ***Start-LBA der Partition***. Sie gibt die LBA-Adresse des ersten Blocks der Partition an. Analog hierzu gibt die darauf folgende ***End-LBA der Partition*** den letzten Block der Partition an. Des Weiteren sind im Partitionseintrag diverse Attribute gespeichert. Diese sind in Tabelle 6.9 auf Seite → separat gelistet. Jeder Partitionseintrag hat einen von Menschen lesbaren Namen, den ***Partitionsnamen***. Hierbei handelt es sich um einen Null-terminierten UCS-2 encodierten String mit einer maximalen Länge von 72 Bytes (Unified EFI, Inc., 2013, 28.2.6 Strings). Die restlichen Bytes eines logischen Blocks werden nicht verwendet und sind ausgenullt.

Partitionstyp- GUIDs

Mit dem Partitionstyp-GUID werden Partitionstypen eindeutig identifiziert. Jeder Betriebssystem-Hersteller muss sich seine(n) eigene(n) Partitionstyp-GUID(s) definieren (Unified EFI, Inc., 2013, 5.3.3 GPT Partition Entry Array). Eine Auswahl an existierenden Partitionstyp-GUIDs ist in Tabelle 6.8 gelistet.

Beschreibung	GUID Value
Unbenutzt	00000000-0000-0000-0000-000000000000
EFI System Partition	C12A7328-F81F-11D2-BA4B-00A0C93EC93B
MBR	024DEE41-33E7-11D3-9D69-0008C781F39F
BIOS Boot Partition	21686148-6449-6E6F-744E-6565644546491
Microsoft Reservierte Partition (MSR)	E3C9E316-0B5C-4DB8-817D-F92DF00215AE
Microsoft Basis Daten Partition	EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
Microsoft Logischer Disk Manager (LDM) Metadaten Partition	5808C8AA-7E8F-42E0-85D2-E1E90434CFB3
Microsoft LDM Daten Partition	AF9B60A0-1431-4F62-BC68-3311714A69AD
Windows Wiederherstellungspartition	DE94BBA4-06D1-4D40-A16A-BFD50179D6AC
Linux Dateisystem	0FC63DAF-8483-4772-8E79-3D69D8477DE4
Linux Logical Volume Manager (LVM) Partition	E6D6D379-F507-44C2-A23C-238F2A3DF928
Linux /home partition	933AC7E1-2EB4-4F13-B844-0E14E2AEF915

[1] „HahIdontNeedEFI“

Tabelle 6.8: Auswahl an GPT Partitionstypen (Unified EFI, Inc., 2013, Table 19. Defined GPT Partition Entry - Partition Type GUIDs).

Attributes

Mit dem Attributfeld können Werkzeuge die genauere Verwendung der jeweiligen Partition angeben. Mögliche Werte sind in Tabelle 6.9 auf der nächsten Seite definiert. Bits 48 bis 63 sind für die jeweilige Benutzung durch die GUID-festlegende Entität reserviert und dürfen auch nur von dieser verändert werden.

Bits	Name	Description
0	Notwendige Partition	Dieses Bit markiert die Partition als unerlässlich für die Funktionalität des Systems.
1	No Block IO Protocol	Es werden keine File System Mappings für diese Partition in UEFI angelegt.
2	PC-AT BIOS Bootbar	Falls dieses Bit gesetzt ist, kann die Partition auch von einem alten PC-AT BIOS gebootet werden.
3-47	Undefiniert	Reserviert (ausgenullt)
48-63	Reserviert für GUID-spezifische Benutzung	Die Benutzung hängt ab von der GUID der Partition.

Tabelle 6.9: GPT Attribute (Unified EFI, Inc., 2013, Table 20. Defined GPT Partition Entry - Attributes).

6.5 Forensischer Zugriff auf Datenträgerdaten

Datenträger enthalten digitale Spuren, die im Rahmen einer forensischen Untersuchung relevant sein können. Beispiele für deratige Spuren sind die Inhalte der Partitionstabelle, die über den weiteren Inhalt der Festplatte Aufschluss geben. Da diese Daten nicht mit bloßem Auge erkennbar sind,

benötigt man Werkzeuge und Methoden, um auf sie zuzugreifen. Dies muss in einer „forensisch sauberen“ Art und Weise geschehen. Das bedeutet, dass die Spuren nicht modifiziert bzw. etwaige Modifikationen dokumentiert werden.

6.5.1 Arten des Zugriffs

Es gibt je nach Situation zwei verschiedene Arten des Zugriffs auf einen Datenträger im Rahmen einer forensischen Untersuchung. Bei der ersten Art erfolgt der Zugriff auf ein „lebendiges“ System. Hierbei ist der Datenträger aktuell im Gebrauch durch einen oder mehrere Computer. Typischerweise erfolgt der Zugriff auf das System über bzw. mit Hilfe des Betriebssystems des laufenden Computers, der die Festplatte benutzt. In diesem Fall besteht das Risiko darin, digitale Spuren zu verändern, d. h. das Betriebssystem und die zu sichernden Daten zu modifizieren. Außerdem besteht die Möglichkeit, dass ein Rootkit den Datensicherungsprozess stört.

Bei der zweiten Art erfolgt der Zugriff auf ein „totes“ System. In diesem Fall liegt der Datenträger (wenigstens kurzzeitig) „in ausgeschaltetem Zustand“ vor, etwa als ausgebaute Festplatte. Der wesentliche Unterschied zum eben genannten Fall liegt darin, dass der Zugriff auf die Daten ohne die Hilfe des Betriebssystems auf dem verdächtigen Rechner geschieht. Dies schließt auch den Fall ein, dass man die Hardware selbst nutzt (wenn man beispielsweise von einer CD oder Floppy bootet). Dies ist der Fall, den wir im Folgenden weiter betrachten wollen.

6.5.2 Schutz der Originaldaten

Beim Zugriff auf die Daten sollten diese nicht verändert werden. Der reguläre Zugriff auf einen Datenträger durch das Betriebssystem (etwa das „***Mounten***“ unter Linux) führt standardmäßig zu Veränderungen. Es ist darum notwendig, den Zugriff mit speziellen Werkzeugen zu unterstützen.

Hardware Write Blocker sind Geräte, die zwischen den Festplatten-Controller und die Festplatte geklemmt werden und schreibende Kommandos auf die Festplatte technisch unterdrücken.

Software Write Blocker versuchen dieselbe Aufgabe zu erfüllen wie Hardware Write Blocker. Software Write Blocker sind entweder Zusatzsoftware zu existierenden Betriebssystemen oder sind eigenständig bootbar. Sie senden nur ungefährliche Befehle zum Datenträger. Dies wird durch „Umbiegen“

(***hooking***) der Unterbrechungsbehandlungen erreicht. Software Write Blocker sind nicht so effizient wie Hardware Write Blocker.

Um sicher zu gehen, dass man keine Daten einer Festplatte übersehen hat, muss auf die Existenz einer HPA oder eines DCO geprüft werden. Um auf die HPA oder DCO zuzugreifen, muss man jedoch die Konfiguration der Festplatte ändern. Diese Datenmodifikation muss dokumentiert werden. Der verwendete (Hardware) Write Blocker muss dies zudem unterstützen. Zur Dokumentation des Zustandes vor dem Sichern schlägt Carrier (2005) vor, erst eine vollständige 1:1-Kopie (siehe unten) ohne HPA und DCO zu machen, anschließend die Konfiguration des Datenträgers zu ändern, und daraufhin eine neue Kopie ***mit*** HPA und DCO zu machen. Ebenfalls sollte der ganze Prozess dokumentiert werden. Das Anfertigen einer vollständigen Kopie kann aber aus verschiedenen Gründen unverhältnismäßig sein. Dies muss im Einzelfall abgewogen und dokumentiert werden.

6.5.3 Fehlerbehandlung

Falls die Festplatte nicht richtig funktioniert, können bestimmte Blöcke nicht gelesen werden. Dies führt zu Fehlern beim Lesen. Die allgemeine Vorgehensweise hierbei ist, den Ort des Lesefehlers zu dokumentieren und als Rückgabewert den Wert 0 zu speichern. Dies erhält die relative Position der restlichen Daten zueinander. Ein weiterer Vorteil dieses Vorgehens ist, dass Controller meist standardmäßig Nullen bei einem Lesefehler lesen.

6.5.4 Abstraktionsstufe und Granularität des Zugriffs

Die Abstraktionsschichten von Computersystemen erlauben verschiedene Ansatzpunkte für die Datensicherung:

- Auf Ebene des Laufwerks,
- auf Ebene der Partitionen,
- auf Ebene der Dateisysteme, oder
- auf Applikationsebene.

Wir konzentrieren uns auf Laufwerke und Partitionen.[7]

Beim Zugriff muss man beachten, dass auf jeder Abstraktionsstufe Daten verloren gehen. Beispielsweise verliert man beim Zugriff auf Partitionsebene die Informationen der Partitionstabelle. Man sollte daher mindestens auf der Ebene zugreifen, welche die interessanten Daten enthält, also „so tief wie nötig“. Andererseits muss der Zugriff auch verhältnismäßig sein, d. h. der Zugriff muss „so hoch wie möglich“ erfolgen.

Man kann beim Zugriff auf die Daten jeweils Datenobjekte unterschiedlicher Granularität sichern. Die Granularität entspricht im Wesentlichen der Größe der zu sicherenden Daten. Das bisherige Standardvorgehen besteht beispielsweise darin, alle Sektoren einer Festplatte zu betrachten und auf ein Sicherungsmedium zu kopieren (sogenannte ***1:1-Kopie*** bzw. ***Festplattenabbild, disk image***). Dies entspricht einem Sichern mit maximaler(gröbster) Granularität. Aber man kann beispielsweise auch nur ein einzelnes Bit vom Datenträger sichern. Dies entspricht in gewissem Sinne einer minimalen (feinsten) Granularität.

Ein Verfahren, das es ermöglicht „so hoch wie möglich aber so tief wie nötig“ und in unterschiedlicher Granularität zu sichern, heißt ***ausgewähltes Sichern*** (auch ***selektives Sichern*** genannt, ***selective imaging***) (Stüttgen u.a., 2013); hierbei wird vereinfach gesprochen nicht die gesamte Festplatte gesichert, sondern nur relevante Teilbereiche, sowie alle relevanten Metadaten der darunterliegenden Schichten. Soll eine selektive Sicherung angestrebt werden empfiehlt sich in jedem Fall die vollständige Lektüre von Stüttgen u.a. (2013).

Früher bestand das Sicherungsmedium aus einer dedizierten eigenen Festplatte, die dann als ***clone copy*** bezeichnet wurde. Die Festplatte wurde vorher „ausgenullt“, um eine Verunreinigung der Daten zu vermeiden. Das Problem bei einer ***clone copy*** ist das Erkennen des Endes der Kopie auf der Zielplatte. Ebenfalls kann es Probleme mit unterschiedlicher Festplattengeometrie geben.

Heute werden Kopien der Daten in Dateien abgelegt. Dies hat den Vorteil, dass man das Ende der Kopie am Ende der Abbild-Datei erkennt.

Für Abbild-Dateien gibt es unterschiedliche Formate:

- ***raw image***: Eine 1:1-Kopie der Festplatte in einer Datei. Solche Abbilder können komprimiert werden (z.B. mit gzip), um Platz zu sparen.
- ***embedded image***: Dies ist ein Containerformat, das typischerweise ein ***raw image*** enthält, zusammen mit zusätzlichen Metadaten wie Hashwerte,

Zeitstempel und Kommentaren.

Zusätzliche Metadaten zu einem Abbild können auch in einer separaten Datei abgelegt werden. Es existieren viele proprietäre Formate für Abbild-Dateien. Ein quell-offenes Format, speziell entwickelt für die forensische Anwendung, ist das Advance Forensic Format (AFF) (Garfinkel u.a., 2006; Garfinkel, 2006).

Sichern über das Netz

Man kann Daten auch über das Netzwerk kopieren. Dies ist besonders dann zu empfehlen, wenn man nicht an die Festplatte herankommt, oder wenn man nicht den richtigen Adapter dabei hat. Dabei wird der Rechner mit einer vertrauenswürdigen CD-ROM gebootet und die Festplatte mit einem Tool über das Netz kopiert. Dies geht zum Beispiel mit dd über ssh.

6.5.5 Wahrung der Integrität

Beim Kopieren der Daten dürfen keine Veränderungen passieren (Wahrung der Integrität der digitalen Spur). Das Bemühen hierzu muss dokumentiert werden. Obwohl es in der digitalen Welt möglich ist, die Integrität einer Kopie mit einem bitweisen Vergleich sicherzustellen, wird aus Effizienzgründen in der Praxis fast ausschließlich ein Hashwertvergleich (Prüfsummenvergleich) eingesetzt. Die Nutzung von Hashwerten eignet sich auch besser für die Dokumentation.

Eine ***Hashfunktion*** ist eine mathematische Funktion, die eine beliebig lange Bitfolge ***m*** auf einen Wert ***h*** mit fester Länge ***n*** abbildet. Mathematisch entspricht das der folgenden Abbildung

$$H : \{0,1\}^* \mapsto \{0,1\}^n$$

und lässt sich als

$$\boldsymbol{h} = \boldsymbol{H}(\boldsymbol{m})$$

schreiben.

Zur Berechnung von Hashwerten kommen ausschließlich sogenannte ***kryptographische Hashfunktionen*** zum Einsatz. Diese müssen die folgenden Eigenschaften haben:

- Einwegfunktion: Es ist praktisch unmöglich, für einen gegebenen Hashwert $\boldsymbol{h}$ die ursprüngliche Bitfolge m zu finden, für die $\boldsymbol{H}(\boldsymbol{m}) = \boldsymbol{h}$ gilt.
- Schwache Kollisionsresistenz: Es ist praktisch unmöglich, für einen gegebenen Hashwert ***h irgendeine beliebige*** andere Bitfolge $\boldsymbol{m'}$ zu finden, für die $\boldsymbol{H}(\boldsymbol{m}) = \boldsymbol{H}(\boldsymbol{m'})$ gilt.

Aus diesen beiden Eigenschaften lässt sich folgern, dass es praktisch unmöglich ist, eine Kopie zu manipulieren, ohne dass sich der Hashwert ändert.

Eine weitere Eigenschaft von kryptographischen Hashfunktionen ist wichtig im Kontext von Integritätssicherung von beliebigen Dateien:

- Starke Kollisionsresistenz: Es ist praktisch unmöglich, zwei ***beliebige*** Bitfolgen $\boldsymbol{m}$ und $\boldsymbol{m'}$ mit $\boldsymbol{H}(\boldsymbol{m}) = \boldsymbol{H}(\boldsymbol{m'})$ zu finden.

Während diese Eigenschaft bei der Integritätssicherung der Kopie eher keine tragende Rolle spielt, ist sie wichtig z.B. beim Festhalten der Integrität von digitalen Dokumenten wie etwa Arbeitsprotokollen. Wäre es z.B. praktisch möglich, zwei Protokolle zu erzeugen, die den gleichen Hashwert, aber unterschiedlichen Inhalt haben, dann könnte ein Ermittler z.B. zwei Übergabeprotokolle erstellen. In einem der Protokolle ist festgehalten, dass ein einziger Datenträger übergeben wurde. Im anderen Protokoll ist festgehalten, dass zehn Datenträger übergeben wurden. Der Ermittler erhält daraufhin 10 Datenträger und bestätigt dies durch digitales Signieren des Hashes des Übergabeprotokolls. Dieses Übergabeprotokoll könnte jetzt aber durch das Protokoll, welches nur die Übergabe eines Datenträgers protokolliert hat, ausgetauscht werden, da die Hashwerte identisch sind. Starke Kollisionsresistenz verhindert jedoch dieses Szenario.

Exkurs 16 (Digitale Signaturen) ***Um eine Nachricht digital zu signieren, kann man ein asymmetrisches Verschlüsselungsverfahren verwenden. Bei einem asymmetrischen Verschlüsselungsverfahren existiert ein öffentlicher Schlüssel e und ein privater Schlüssel d. Der öffentliche Schlüssel (public key) ist öffentlich bekannt, der private (private key oder secret key) ist geheim und nur seinem Besitzer bekannt. Mittels des öffentlichen Schlüssels kann nun eine Nachricht p als*** $\boldsymbol{c = E_e(p)}$ ***verschlüsselt werden. Die verschlüsselte Nachricht c kann dann mit dem privaten Schlüssel als*** $\boldsymbol{p = E_d(c)}$ ***entschlüsselt***

werden. Dabei ist wichtig, dass man einerseits weder d aus e noch andererseits p aus c oder c aus p ohne Kenntnis von d berechnen kann.

Ebenfalls aus Effizienzgründen verwendet man zum Signieren nicht das gesamte Dokument, sondern nur den kryptographischen Hash h des Dokuments m. Der Hashwert h wird berechnet als $h = H(m)$.

Für E gilt ebenfalls $E_e(E_d(x)) = E_d(E_e(x))$.

Beim Signieren wird dann die Signatur s als $s = E_d(h)$ ***berechnet. Zum Verifizieren der Signatur berechnet man den Hash*** $h' = H(m')$ ***des zu verifizierenden Dokuments m′ und prüft*** $h' \stackrel{!}{=} E_e(s)$. ***Stimmt dies überein, also gilt*** $E_e((E_d(H(m))) = E_d\{E_e\{H\{m'\}\})$, ***so ist die Signatur gültig. Da nur der Besitzer des privaten Schlüssels e dessen Wert kennt, und man (siehe oben) s ohne Kenntnis von e nicht berechnen kann, ist ausgeschlossen, dass die Signatur von einer anderen Person erstellt wurde***.

Problem mit Embedded Images

Embedded Images speichern den Hashwerte in derselben Datei wie das Image. Dadurch können Image und Hashwert gleichermaßen modifiziert werden. Daher muss man dokumentieren, welchen Hashwert das Image zu welcher Zeit hatte. Hierzu ist der Nutzen eines handgeschriebenen Logbuches ratsam. Ein Angreifer müsste somit nicht nur die Dateien ändern, sondern auch das Logbuch.

Teil-Hashwerte

Die Eigenschaften von kryptographischen Hashfunktionen führen dazu, dass sich der Hashwert eines Bitstings komplett ändert, wenn nur ein einzelnes Bit im Original verändert wurde. Eine solche Veränderung kann durch Alterung von Daten auf Festplatten jedoch leicht passieren. Dies hätte einen kompletten Zusammenbruch der Beweiskette zur Folge. Um dies zu vermeiden, kann man alternativ einzelne Hashwerte für Teile der Daten (***chunks***) berechnen. Der Verlust der Daten beschränkt sich dann nur auf den jeweiligen chunk, dessen Hashwert nicht mehr stimmt. Chunks können sowohl kontextfrei, d. h. mit fester n Byte-Länge definiert werden, aber eine kontextabhängige Definition ist ebenfalls möglich (Kornblum, 2006).

Spezialhardware

Einige Hersteller bieten „all in one“ Lösungen für eine Datensicherung an. Das sind meistens Spezialcomputer mit Software, Kabeln und Schnittstellen, die in einem praktischen Koffer für den Einsatz vor Ort arrangiert sind. Diese Geräte können meist mit hoher Datenrate kopieren und berechnen gleichzeitig die kryptographischen Hashwerte. Einige Beispiele sind:

- Image MaSSter Solo 101 Forensic[8]
- TreCorder[9]

6.6 Zusammenfassung

In diesem Kapitel haben wir die Methodik der forensischen Informatik am Beispiel von Partitionssystemen dargestellt. Die Fragen waren: Wie geht man mit digitalen Spuren um und was bedeuten diese Spuren?

Bei der Betrachtung des Zugriffs auf die digitalen Spuren haben wir auch grundsätzliche Fragen besprochen, die im Rahmen einer allgemeinen Datenträgeranalyse aufkommen. Während dort die „klassische“ 1:1-Sicherung noch von großer Relevanz ist, benötigt man zur Analyse der Partitionsstrukturen in der Regel nur den (lesenden) Zugriff auf die Partitionstabelle, also in der Regel die ersten Sektoren des Laufwerks. Die Analyse der ***in*** den Partitionen gespeicherten Daten und ihrer Organisation (Dateisysteme) wird andernorts (Carrier, 2005) vertieft behandelt.

---

[6]Wie später noch erläutert wird, darf man diesen String nicht verwenden, um den Typ des Dateisystems abzulesen, also ob es ein FAT32, FAT16 oder FAT12-Dateisystem ist. Die Strings können aber zur Lokalisierung des Dateisystems bzw. der Partition verwendet werden.

[7] Prinzipiell könnte man statt der auf der Festplatte gespeicherten Bits auch die Magnetisierung der Oberfläche messen und abspeichern. Dies wären dann aber keine digitalen Spuren mehr.

[8]http://ics-iq.com/image-masster-solo-102-g3-forensic-hard-drive-data-acquisition-unit

[9]http://mh-service.de/index.php/de/produkte/mh-systeme/trecorder

Kapitel 7

Dokumentation

Autoren: Andreas Dewald, Felix Freiling

Die Sicherung und Analyse von Spuren muss jederzeit nachvollziehbar sein, unter Umständen auch noch Jahre nachdem die Spuren aufgenommen wurden. Die Dokumentation des Vorgehens und der Ergebnisse einer Untersuchung spielt also eine zentrale Rolle im Rahmen einer digitalen Ermittlung. Dies gilt nicht nur für den unabhängigen Sachverständigen, der die Ergebnisse einer Ermittlung im Rahmen eines Gerichtsverfahrens kritisch prüft, sondern auch für den digitalen Ermittler selbst, der häufig vor Gericht zu einer mehrere Jahre zurückliegenden Ermittlung befragt wird.

Es gibt aus der klassischen Forensik verschiedene Grundsätze, die auch darauf abzielen, die Nachvollziehbarkeit zu erleichtern. Ein solcher Grundsatz fordert, die gefundenen Spuren nicht zu verändern. Unter bestimmten Bedingungen können jedoch Veränderungen der Spuren nicht ausgeschlossen werden – auch bei digitalen Spuren. Beispielsweise werden bei der Live-Analyse eines zu untersuchenden Systems viele Zeitstempel und Hauptspeicherinhalte verändert. Gerade in solchen Situationen muss man genauestens alle Schritte dokumentieren, um auch diese Änderungen immer und überall nachvollziehbar zu machen. Dazu gehört auch immer eine Begründung, warum diese Veränderungen notwendigerweise geschehen mussten. Oft ist es in solchen Fällen schwer, überhaupt alle Änderungen zu überblicken. Insofern bedeutet das Wissen, dass nichts verändert wurde, lediglich, dass man nichts bewusst verändert hat.

Dieses Kapitel betrachtet ausgewählte Aspekte der Dokumentation von digitalen Ermittlungen. Nach ein paar allgemeinen Punkten zur Integrität solcher Berichte entwickeln wir einen Vorschlag für deren Struktur und geben

verschiedene positive und negative Beispiele aus Berichten, die Studierende im Rahmen von Lehrveranstaltungen angefertigt haben.

7.1 Allgemeine Aspekte

Insgesamt lautet das Motto bei forensischen Untersuchungen: Dokumentieren, Dokumentieren, und nochmals Dokumentieren! Wir betrachten zunächst eine Reihe allgemeiner Aspekte, die an Erfahrungen aus der klassischen Forensik anknüpfen.

7.1.1 Verwahrungskette (engl. chain of custody)

Wie schon mehrfach erwähnt dokumentiert die ***Verwahrungskette*** (***chain of custody***) lückenlos, wann welche Personen Zugang zur Spur hatte und ebenfalls wann keine Personen Zugang zur Spur hatten, sowie den Aufbewahrungsort während dieser Zeit. Eine gute Verwahrungskette dokumentiert somit den örtlichen und zeitlichen Verlauf und auch den Zustand der Spuren. Sie umfasst unter Anderem Zeit, Ort und durchführende Person der Beschlagnahme, Aufnahme, Übergabe, Analyse und Verbleib der Spur.

Bei digitalen Spuren bezieht sich die Dokumentation der Verwahrungskette in der Regel auf den physischen ***Spurenträger*** (also den Datenträger). Dies korrespondiert mit dem klassischen Bild des Asservates, das in der Asservatenkammer aufbewahrt wird. Man dokumentiert dann beispielsweise, wer wann darauf Zugriff hatte und wann es wieder „weggeschlossen“ wurde. Digitale Spuren werden aber häufig dupliziert und anschließend auf großen Datenspeichern im Netzwerk vorgehalten. Auch hier muss sichergestellt (und dokumentiert) werden, wer wann Zugriff auf die Spur hatte und was damit geschah. Zur Dokumentation gehören also auch Zugriffsprotokolle, die durch die Speichersysteme verlässlich generiert werden müssen.

7.1.2 Handschriftliche Dokumentation

Gerade bei digitalen Ermittlungen sollten klassische Dokumentationstechniken nicht vergessen werden. So sollte während der Untersuchung ein handschriftliches Logbuch geführt werden, in das wesentliche Arbeitsschritte und wichtige Daten (Uhrzeiten, Passwörter, Prüfsummen) eingetragen werden.

Die Notizen sollten möglichst auf nummerierten und gebundenen Seiten gemacht werden. Jede wichtige Angabe und Schlussfolgerung kann mit dem eigenen Namenskürzel signiert und mit Ort und Zeit der Aufzeichnung versehen werden. All diese Maßnahmen erhöhen den Beweiswert der Daten, da eine nachträgliche Manipulation bei derartig erstellten Notizen schwerer ist als bei rein digital erzeugter Dokumentation. Außerdem erlaubt die Handschrift (mit wiederholten Namenskürzeln) auch die direkte Zuordnung der Aufzeichnungen zur ermittelnden Person.

Wann immer möglich, sollte bei einer Untersuchung das Vier-Augen-Prinzip angewendet werden. Gerade in Situationen, bei denen andere Aufzeichnungsmöglichkeiten nicht verfügbar sind, also etwa bei der Live-Analyse, lohnt es sich also, eine weitere fachkundige Person hinzuzuziehen, die handschriftlich (im Logbuch) ein bestimmtes Vorgehen bestätigt. Diese Person kann später als Zeuge geladen werden, um so die Glaubwürdigkeit der Untersuchung zu unterstützen.

7.1.3 Automatische Dokumentation

Nicht alle Schritte müssen (und können) bei digitalen Ermittlungen handschriftlich festgehalten werden. Für eine vollständige und damit nachvollziehbare Dokumentation ist es aber trotzdem erforderlich, jeden Schritt zu dokumentieren – wirklich jeden. Um dies bei der Analyse von digitalen Spuren zu vereinfachen, gibt es Hilfsmittel für die automatische Dokumentation. Viele kommerzielle Werkzeuge erlauben es, den Fortgang der Untersuchung durch Lesezeichen, Kommentare oder andere Informationen anzureichern. Die wesentlichen Aktivitäten an der Benutzerschnittstelle werden standardmäßig festgehalten und in das am Ende automatisch erstellte Protokoll übernommen.

In besonderen Situationen, etwa im Kontext einer Live-Analyse, ist es hilfreich, die Befehle (und ihre Ausgaben) zu dokumentieren, die auf der Kommandozeile eingegeben werden. Hierzu wird häufig das Unix-Programm `script` benutzt. Dieses Programm wurde jedoch nicht zum Zwecke forensischer Untersuchungen geschrieben und hat verschiedene Unzulänglichkeiten etwa bei der Speicherung von Dateiausgaben, die Steuersequenzen enthalten. Inzwischen existiert mit `forscript` (forensic script) eine Alternative, die problemlos auch bei digitalen Ermittlungen eingesetzt werden kann (Dewald u.a., 2010).

Da in der Praxis unter Windows viele Interaktionen über die grafische Schnittstelle und nicht auf der Kommandozeile erfolgen, ist die automatische

Dokumentation von Aktivitäten dort häufig schwieriger als unter Linux. Eine Möglichkeit besteht darin, das eigene Vorgehen per Videomitschnitt des Bildschirms oder durch Bildschirmfotos zu dokumentieren. Hierfür gibt es neben der Standardmöglichkeit, Bildschirmkopien zu erzeugen, noch zahlreiche andere zum Teil kommerzielle Werkzeuge, insbesondere für den Videomitschnitt des Bildschirms, etwa Camtasia (TechSmith Corporation, 2011), ScreenKast (Beligum, 2009) oder recordMyDesktop (Varouhakis, 2008). Die genannten Programme müssen auf dem System verfügbar sein. Alternativ gibt es auch Geräte, mit denen man das Signal am VGA-Ausgang eines Rechners mitschneiden kann (Epiphan Systems Inc., 2011).

Automatisch generierte Dokumentationen, wie Texte oder Bilder, sind wieder Daten, die prinzipiell leicht manipulierbar sind. Zur Sicherung der Integrität dieser Dokumente kann man die kryptographischen Prüfsummen (die Hashwerte) der Dateien berechnen lassen und diese dann handschriftlich in das Logbuch übernehmen. Bei Zweifeln an der Integrität kann später der Hashwert der fraglichen Datei mit dem im Logbuch verzeichneten Wert verglichen werden. Sind beide identisch, kann eine Manipulation mit sehr großer Wahrscheinlichkeit ausgeschlossen werden.

7.1.4 Dokumentation von Zeit

Die Dokumentation von Zeit spielt bei jeder forensischen Untersuchung eine große Rolle. Das betrifft einerseits die Zeitpunkte der Ereignisse, die den Tathergang ausmachen (siehe Kapitel 1), andererseits auch die Zeitpunkte, an denen der Ermittler bestimmte Aktivitäten durchgeführt hat (Auswertungszeit).

Die Dokumentation von Zeit bei digitalen Spuren basiert in der Regel auf Zeitstempeln, die Computer hinterlassen. Zeitstempel sind eine spezifische Eigenart digitaler Spuren, gerade innerhalb von Dateisystemen. Die Interpretation von Zeitstempeln ist aber aus vielfältigen Gründen schwierig. Ein wesentlicher Grund ist, dass der Zeitpunkt, der durch den Zeitstempel dokumentiert wird, nicht der Zeitpunkt sein muss, zu dem der Zeitstempel gesetzt worden ist. Bei der Rekonstruktion von Ereignissen muss man demnach mindestens zwei verschiedene Zeiten unterschieden:

1. Die ***Zeit des Untersuchungsobjekts***, also beispielsweise den Wert der internen Uhr eines beschlagnahmten Rechners. Diese Zeit ist in der Regel ausschlaggebend für das Setzen von Zeitstempeln auf dem System selbst.

2. Die „echte“ Zeit, also die Zeitpunkte, zu denen die Ereignisse tatsächlich stattfanden.

Beide Zeiten müssen nicht notwendigerweise übereinstimmen (siehe Abbildung 7.1).

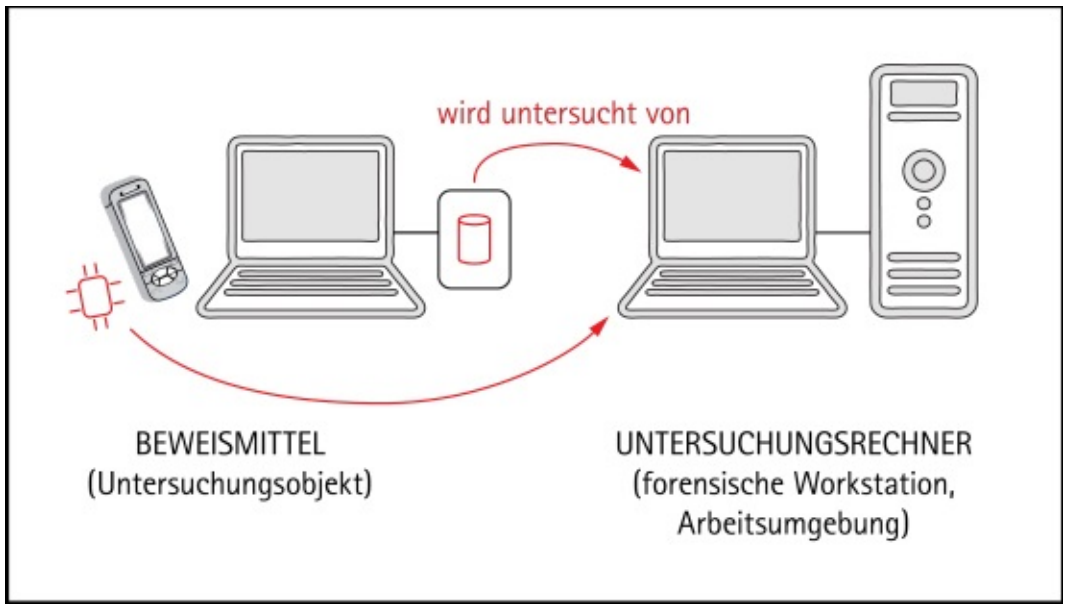


Abbildung 7.1: Untersuchter Rechner und Untersuchungsrechner.

Es ist also sehr wichtig, die Zeit auf dem untersuchten System festzustellen und zu dokumentieren. Dies kann beispielsweise im Rahmen einer Live-Analyse geschehen. Ist der Rechner bereits ausgeschaltet, kann man ihn mittels einer Live-CD kurzzeitig hochfahren, um den Wert der internen Uhr auszulesen. Auch dies muss dokumentiert werden.

Weitere Gründe, die die Interpretation von Zeitstempeln erschweren, sind die vielen unterschiedlichen Zeitformate, in denen Zeitstempel in Dateisystemen abgelegt werden. Auch wird die Zeitzone oft nicht mit abgespeichert. Auch die Systemzeit muss sich nicht immer gleich von der echten Zeit unterscheiden, da viele Betriebssysteme ihre interne Uhr mit Uhrzeitservern im Internet synchronisieren, sobald sie mit dem Netz verbunden sind. Das Vorhandensein von Zeitstempeln sollte also nicht dazu verleiten, die Zeitpunkte von Ereignisse mit wissenschaftlicher Exaktheit auf einer absoluten Zeitskala einzuordnen. Wie in der klassischen Forensik kann der Zeitpunkt eines Ereignisses auch in der digitalen Forensik bestenfalls eingegrenzt werden. Je weiter die fraglichen Zeitpunkte zurückliegen, desto schwieriger wird deren zeitliche Eingrenzung.

Um die Nachvollziehbarkeit einer automatisch generierten Dokumentation zu erleichtern, sollten auch die Zeitpunkte dokumentiert werden, an denen bestimmte Untersuchungsaktionen durchgeführt wurden. Am einfachsten geschieht dies durch die Ausgabe der lokalen Zeit des Untersuchungsrechners in die automatisch generierte Dokumentation. Dazu muss die Bezugszeit auf dem Untersuchungsrechner aber immer korrekt sein (***Coordinated Universal Time***, UTC). Man sollte also immer für eine korrekte Zeit am eigenen Arbeitsplatz sorgen, etwa mittels Uhrensynchronisationsprotokollen wie NTP und dies auch dokumentieren.

7.1.5 Vorgeschriebene Mindestanforderungen

Das Sachverständigenwesen ist im juristischen Kontext umfangreich etabliert und zum Teil auch genau reguliert. Für einige Sachgebiete haben etwa die Industrie- und Handelskammern Mindestanforderungen an Gutachten herausgegeben, die den fachlichen Standard festschreiben und die Sorgfaltspflichten des Sachverständigen in fachlicher Hinsicht konkretisieren. Das Wissen um diese Regelungen ist darum notwendig, um als Sachverständiger auftreten zu dürfen. Dieses Wissen wird in der Regel bei der Sachverständigenausbildung durch die entsprechenden Berufsverbände vermittelt.

7.2 Nachvollziehbarkeit

7.2.1 Nachvollziehbarkeit vs. Nachprüfbarkeit

Wie eingangs erwähnt, ist die Nachvollziehbarkeit der Analyse eines der wichtigsten Qualitätskriterien forensischer Dokumentation. Nachvollziehbarkeit bezieht sich auf die Plausibilität der dokumentierten Untersuchungsschritte. Hilfreich ist dabei insbesondere auch die Begründung, warum etwas (in einer bestimmten Reihenfolge) gemacht wurde und gegebenenfalls auch warum etwas ***nicht*** gemacht wurde. Schlussfolgerungen sollten dadurch möglichst unabhängig vom Betrachter nachvollzogen werden können. Um dies bereits bei der Anlage der Dokumentation zu begünstigen, sollte man auch die Datenbasis dokumentieren, auf der die eigenen Schlüsse basieren. Sachverständige unterscheiden darum zwischen Nachvollziehbarkeit und Nachprüfbarkeit:

Nachprüfbarkeit bezieht sich auf die unabhängige Nachvollziehbarkeit von Untersuchungsergebnissen ausgehend von verfügbaren Basisdaten.

7.2.2 Verhältnismäßigkeit der Dokumentation

Die reine Lehre der Forensik schreibt vor, jeden Untersuchungsschritt zu begründen und zu dokumentieren. Dies sollte vor allem auch in der Ausbildung im Vordergrund stehen. Allerdings ist eine derart umfangreiche Dokumentation in der Praxis häufig weder notwendig noch angebracht.

Beispiel 34 (Dokumentation der Existenz einer Datei) ***Wenn das Ermittlungsziel lautet, die Existenz einer Datei auf einem Datenträger zu überprüfen, dann reicht es im positiven Fall in der Regel aus, den Fundort der Datei anzugeben (Dateiname und -pfad, ggf. Sektornummer, etc.). Solange die Originalspur noch unverändert vorliegt, kann mit Hilfe der Angabe des Fundorts die Frage nach der Existenz der Datei schnell und unabhängig geprüft werden. Zur Prüfung ist es dann nicht notwendig im Detail nachzuvollziehen, auf welche Weise die Datei gefunden wurde***.

In der Praxis sind Untersuchungen auch an einen Kostenrahmen gebunden, der es verbietet, mehr als nur einen minimalen Dokumentationsaufwand zu investieren. So beschränkt sich die Dokumentation häufig auf die eigentlichen Funde, sowie allerlei automatisch generierte Protokolle, die mit gelegentlichen Notizen angereichert sind. Im deutschen Rechtskontext ist zudem von Relevanz, dass Fehler beim Vorgehen nicht notwendigerweise dazu führen, dass Funde nicht vor Gericht verwertet werden dürfen. Das strafrechtliche Verwertbarkeitsverbot beschränkt sich in Deutschland auf relativ wenige Bereiche (beispielsweise Erlangung von Beweismitteln durch Folter), so dass der Richter in der Würdigung der vorgebrachten Spuren relativ frei ist. Dies steht im Gegensatz zur US-amerikanischen Rechtstradition, die auch die Verwertung von Beweismitteln verbietet, die auf Basis von unrechtsmäßig erlangten Beweismitteln gefunden wurden (***fruit of the poisonous tree***).

7.2.3 Versionierung

Zur guten Nachvollziehbarkeit gehört auch, die Arbeit am Bericht selbst zu dokumentieren, also den aktuellen Stand des Berichts mit einem Datum zu

versehen („Stand: 29.4.2010“) oder Versionsnummern zu vergeben. So können unterschiedliche Versionen eines Dokuments leichter auseinander gehalten werden. Dabei ist es sinnvoll, das Datum oder die Versionsnummer in der Kopf- oder Fußzeile des Berichts aufzuführen. Bei mehreren Versionen sollte der Bericht zusätzlich eine vollständige Versionshistorie enthalten, die alle vorherigen Versionen mitsamt einer kurzen Beschreibung der am Bericht durchgeführten Änderungen dokumentiert. Bei mehreren Autoren sollte auch kenntlich gemacht werden, wer die Änderungen durchgeführt hat. Die Verwaltung verschiedener Dokumentversionen kann mit Werkzeugen zur Versionsverwaltung wie etwa Subversion (Apache Software Foundation, 2011) oder Git (Git Development Team, 2011) automatisch unterstützt werden.

7.2.4 Beispiele

Die wissenschaftliche Methode (siehe Kapitel 1) verlangt, dass jeder Schluss selbst hinterfragt wird und Vermutungen als solche gekennzeichnet werden. Es folgen abschließend einige Beispiele, die das verdeutlichen sollen.

Beispiel 35 (Vermutungen) ***Betrachten wir den folgenden Satz: „Die Datei wurde durch den Benutzer Meier angelegt“. An diesem Satz ist unklar, ob mit „Benutzer Meier“ eine tatsächliche Person namens Meier gemeint ist oder lediglich die Benutzerkennung mit der Bezeichnung `Meier`. Falls eine konkrete Person gemeint ist, ist zudem fraglich, wie der Zusammenhang zwischen der digitalen Spur und der realen Person hergestellt wurde. Eine präzisere Formulierung wäre hier: „Die Datei wurde durch einen Benutzer mit dem Login-Namen `Meier` angelegt.“***

Beispiel 36 (Juristische Einschätzungen) ***Insbesondere dürfen auch keine juristischen Einschätzungen vorweggenommen werden. Die juristische Einschätzung der Sachlage ist Aufgabe des Richters. Dies trifft vor allem im Kontext von Delikten zu, bei denen es einen großen Auslegungsspielraum gibt. Es wäre beispielsweise falsch zu schreiben: „Datei `X` enthält kinderpornographische Inhalte.“ Da die letztendliche Einstufung einer Datei als Kinderpornographie vom Richter vorgenommen werden muss, formuliert man hier besser: „Datei `X` enthält vermutlich kinderpornographische Inhalte.“ oder „Datei `X` enthält kinderpornographisch verdächtige Inhalte.“***

Beispiel 37 (Nachvollziehbarkeit und Nachprüfbarkeit) ***Die Nachvollziehbarkeit eines Berichts soll auch die leichte Reproduzierbarkeit beziehungsweise die leichte Prüfbarkeit der Ergebnisse fördern. Dazu ist es oft hilfreich, sich in die Lage eines unabhängigen Sachverständigen zu versetzen, der den Bericht lesen und prüfen soll. Die Aussage „Auf der Festplatte wurden die Reste einer Bilddatei gefunden.“ ist zum Beispiel zwar präzise, aber nur sehr schwer überprüfbar. Besser wäre eine Formulierung der Art „In Festplattensektor 325 wurden Reste einer JPG-Datei gefunden.“ Diese Formulierung ermöglicht es beispielsweise, durch das bloße Analysieren des angegebenen Sektors sofort die Bilddatei zu rekonstruieren und die Behauptung schnell zu überprüfen***.

7.3 Aufbau forensischer Berichte

Wir betrachten im Folgenden den Aufbau forensischer Berichte. Dieser Aufbau orientiert sich an der Gestaltung wissenschaftlicher Texte (wie Fachpublikationen oder Seminararbeiten) und fordert gleichzeitig eine entsprechende sprachliche Präzision. In den jeweiligen Teilen sollte man sich zudem auf das Wesentliche konzentrieren.

7.3.1 Zielpublikum

Beim Schreiben eines forensischen Berichtes ist es wichtig, sich über das Zielpublikum bewusst zu sein. Denn primär richtet sich die Dokumentation einer forensischen Untersuchung an Nicht-Techniker und Juristen, wie beispielsweise Staatsanwälte, Richter, Juristen aus der Innenrevision oder aus der Sicherheits- oder Personalabteilung. Selbstverständlich kann es auch sein, dass sich der Vorstand oder die Geschäftsführung einer Firma für die Untersuchungsergebnisse interessiert.

Erst in zweiter Linie richtet sich der Bericht an Spezialisten (Techniker), welche die Ergebnisse überprüfen wollen. Diese beiden Zielgruppen haben in der Regel orthogonale Interessen: Nicht-Techniker verstehen Fachbegriffe nicht und möchten möglichst einfache und kurze Schlussfolgerungen lesen. Techniker hingegen legen Wert darauf, alle Details zu sehen und diese nachvollziehen und unter Umständen auch nachprüfen zu können. Um diesen unterschiedlichen Interessen nachzukommen, ist es sinnvoll, einen forensischen Bericht in verschiedene Abschnitte zu gliedern, die im Folgenden besprochen werden.

7.3.2 Grobgliederung

In diesem Abschnitt möchten wir exemplarisch den groben Aufbau eines forensischen Berichtes vorstellen. Er orientiert sich implizit an der Struktur, die sich im Rahmen der Lehre bei der Untersuchung von Speichermedien (Festplatten) bewährt hat, ist aber relativ allgemein gehalten, so dass er für andere Arten von Spuren auch in variierter Form verwendet werden kann. Eine ähnliche Struktur empfiehlt auch Casey (2011, S. 76ff), auch wenn dessen Struktur stark auf Datenträgeranalysen ausgerichtet ist. Allerdings verlangen unterschiedliche Deliktsbereiche unterschiedliche Dokumentation. Während die hier vorgestellte Gliederung sicherlich für eine einfache Datenträgeranalyse sinnvoll erscheint, so benötigt man für komplexe Analysen (etwa im Umfeld von Wirtschaftskriminalität) angepasste Formate.

Als Erstes muss der Bericht natürlich einen Titel (zum Beispiel „Untersuchungsbericht“, „Abschlussbericht“, „Gutachterliche Stellungnahme“) sowie nähere Angaben zu seiner Entstehung enthalten. Dazu gehören die Namen der Autoren, deren Dienststellen oder Abteilungen, gegebenenfalls auch Verfahrensnummern, Aktenzeichen und die Versionshistorie. Diese Angaben sollen es dem Leser erlauben, den Bericht schnell in einen Verfahrenskontext einzubetten.

Als Nächstes folgt ein Prolog, der dem Leser schnell einen Überblick über den genaueren Kontext und den Umfang der Untersuchung bietet. Hierzu sollten beispielsweise folgende Angaben gemacht werden:

- Eine genaue Auflistung der untersuchten Spuren (zum Beispiel Festplatten-Seriennummern, Asservatennummern, Netzwerkmitschnitte). Zur Dokumentation der Spuren gehört auch eine Beschreibung der ***chain of custody***, also die lückenlose Auflistung der Stellen und Personen, die im relevanten Zeitabschnitt der Untersuchung mit der Spur zu tun hatten.

- Beschreibung des Untersuchungsauftrags, also zum Beispiel der Fallkontext, der vermutete Straftatbestand.

- Beschreibung der Arbeitsumgebung, also wann und wo mit den Spuren gearbeitet wurde und welche Werkzeuge in welcher Version dabei zum Einsatz kamen. Diese Angaben sollen helfen, Zweifel an der Vertrauenswürdigkeit der Arbeitsumgebung auszuräumen. Gegebenenfalls

kann es auch sinnvoll sein, die Expertise der an der Untersuchung beteiligten Personen stichpunktartig zu belegen (ein ausführlicher Lebenslauf kann bei Bedarf im Anhang erscheinen).

Im Anschluss folgt die Zusammenfassung des Berichts für Nicht-Techniker. Dieser Teil richtet sich an Leser, die an den Ergebnissen des Berichts interessiert sind und sich nicht mit technischen Details beschäftigen wollen. Die Formulierungen sollten darum auf technische Fachbegriffe weitestgehend verzichten. Hier soll alles Wesentliche auf maximal einer Seite zusammengefasst werden. Wesentlich sind Schlussfolgerungen aus den untersuchten Spuren, die für den Untersuchungsauftrag relevant sind.

Im nächsten Abschnitt folgt eine etwas ausführlichere Zusammenfassung für Techniker. Das Zielpublikum dieses Teils besteht aus technischen Sachverständigen, die sich für die Details der Untersuchung interessieren und die Untersuchung nachvollziehen wollen. Die Länge dieses Dokumentteils ist natürlich abhängig vom Umfang und vom Verlauf der Untersuchung. Wie in der Zusammenfassung für Nicht-Techniker geht es aber auch hier darum, einem Experten schnell die wesentlichen Schritte der Untersuchung darzulegen. Insofern sollte die Darstellung sich auf etwa zehn Seiten beschränken. Wichtig für diesen Teil ist eine übersichtliche Gliederung, bei der die logische Folge der Untersuchungsschritte sichtbar wird. In diesem Teil kann auch direkt Fachsprache verwendet werden, etwa mit direkter Angabe von Fundstellen (wie Hauptspeicheradressen oder Festplattensektoren) oder dem Bezug auf Untersuchungsschritte, die mit speziellen Werkzeugen durchgeführt wurden.

Natürlich kann man auf zehn Seiten nicht alle Details einer Untersuchung darstellen. Diese Details können in einem abschließenden Teil des Berichts in aller Ausführlichkeit dargelegt werden. Man kann diesen Teil des Berichts als eigenes Kapitel („Vollständige Untersuchungsdokumentation“) oder als Anhang zur Zusammenfassung für Techniker gestalten. In diesem Teil kann man auch längere Listings, Logdateien, oder Sequenzen von Bildschirmfotos zeigen, die bis ins Detail den Untersuchungsverlauf darstellen.

1. Titelseite:
   Name des Autors, Dienststelle, Aktenzeichen, Versionshistorie
2. Prolog:

Liste der Beweismittel, ***chain of custody***, Untersuchungsauftrag, Arbeitsumgebung

3. Zusammenfassung für Nicht-Techniker (maximal eine Seite):
   Aufzählung wesentlicher Ergebnisse in allgemeinverständlicher Sprache
4. Zusammenfassung für Techniker (maximal zehn Seiten):
   Dokumentation der wesentlichen Schritte bei der Ermittlung in Technikersprache (bei Details auf den Anhang verweisen)
5. Details für Techniker (meist als Anhang, beliebig lang):
   Ausführliche Darstellung der Details der Ermittlung (Logdateien, Bildschirmfotos, Listings usw.)

Alle Abschnitte (auch der Anhang) sind durchnummeriert, um in anderen Teilen des Berichtes darauf Bezug nehmen zu können.

Abbildung 7.2: Grobgliederung eines Untersuchungsberichts.

Abbildung 7.2 fasst die wesentlichen Abschnitte der Grobgliederung nochmals in einer Übersicht zusammen.

7.4 Vorgehen bei der Erstellung

Mit der Erstellung des Berichtes sollte bereits zu Beginn der Ermittlung begonnen werden. Folgendes Vorgehen bietet sich an: Zunächst wird eine Sammlung aller gefundenen Spuren und Ermittlungsschritte in einer großen Datei (oder einem Verzeichnis) angelegt. Die Einzelschritte der Untersuchung notiert man handschriftlich im Logbuch. Die Spurensammlung wird später quasi zum Anhang des Berichts. Während der Untersuchung werden alle wesentlichen technischen Schritte für die Zusammenfassung für Techniker extrahiert und ebenso alle wesentlichen Schlussfolgerungen, welche in die Zusammenfassung für Nicht-Techniker einfließen. Der eigentliche Bericht wird dann aus den Informationen des Logbuchs und der Datensammlung vervollständigt.

Textdateien, verwendete Skripte oder relevante Fotos können direkt im Bericht abgedruckt werden. Problematischer ist die Darstellung anderer verfahrensrelevanter Dateiformate (Datenbanken, Videos, Webseiten). Hier

muss darauf geachtet werden, dass die Dateiinhalte möglichst unverfälscht dargestellt werden. Gegebenenfalls ist die Darstellungsform zu dokumentieren.

Insgesamt sollte sich die Vorgehensweise bei der Untersuchung an einem forensischen Vorgehensmodell (siehe Kapitel 5) orientieren. Natürlich muss dann auch dokumentiert werden, welches Modell verwendet wird.

7.5 Beispiele aus forensischen Berichten

In diesem Abschnitt möchten wir einige Auszüge aus forensischen Berichten diskutieren, die von Studierenden im Rahmen der Lehrveranstaltung „Forensische Informatik“ an den Universitäten Mannheim und Erlangen-Nürnberg erstellt wurden. Den Studierenden wurde dabei jeweils ein Festplattenabbild auf einer CD zur forensischen Untersuchung ausgehändigt. Es handelt sich um eigens hierfür erstellte Abbilder rein fiktiver Fälle.

7.5.1 Bericht 1

Abbildung 7.3 auf der nächsten Seite zeigt die Gliederung von Bericht 1, die der zuvor dargestellten Empfehlung folgt: Nach allgemeinen Vorbemerkungen über die Arbeitsumgebung und den Arbeitsauftrag folgt eine kurze, stichpunktartige Zusammenfassung. Anschließend gibt die technische Analyse detailliertere Einblicke in den Fortgang der Untersuchung. Der Anhang sammelt schließlich Protokolldateien, wiederhergestellte Dokumente und Bilder.

Aus dem eigentlichen Text des Berichts greifen wir den Abschnitt I-B heraus. Dieser Teil enthält eine gut nachvollziehbare und vollständige Darstellung der ***chain of custody***. Der studentische Humor ist in realen Berichten natürlich unangebracht.

> I-B. Nachweis über die Integrität des Asservates:
>
> Das Image wurde am 25.03.2010 in der Vorlesung Forenische Informatik auf einer CD-ROM ausgegeben. Das Asservat wurde von zwei Polizeibeamten der Dienststelle CSI Neckarbrücke in das forensische Labor der Dienststelle transportiert. Das Image wurde dort in einem Tresor aufbewahrt und nur zum Kopieren des Images am 06.04.2010 um 16:00 Uhr herausgenommen. Die Integrität wurde zu jedem Zeitpunkt durch Überprüfen eines md5-Hashwertes sichergestellt. Die Untersuchung wurde

am 06.04.2010 von 16:00 Uhr bis 22:00 Uhr und am Folgetag von 12:00 Uhr bis 18:00 Uhr durchgeführt. Die Analyse fand im forensischen Labor der Dienststelle CSI Neckarbrücke in der Arno-Nym-Straße statt. Der Zutritt zu diesem Labor ist nur ausgewählten Mitarbeitern gestattet. Diese wurden umfangreich im Umgang mit Asservaten geschult. Die Räumlichkeiten werden durch ein hochmodernes Schloss gesichert und von Kommissar REX, dem Dienststellenhund, rund um die Uhr bewacht.

INHALTSVERZEICHNIS

I Vorwort 3
- I-A Auftrag . . . 3
- I-B Nachweis über die Integrität des Aservates . . . 3
- I-C Eingesetzte Software . . . 3
- I-D Eingesetzte Hardware . . . 3

II Zusammenfassung 4
- II-A Stichpunktartige Analyse des Endzustands . . . 4
- II-B zeitlicher Ablauf . . . 5
- II-C Vermutungen über den Kontext: . . . 5

III Technische Analyse 6
- III-A Einleitung . . . 6
- III-B Maßnahmen . . . 6
  - III-B1 Kopieren und Entpacken . . . 6
  - III-B2 mmls . . . 6
  - III-B3 Einlesen in Autopsy . . . 7
  - III-B4 Aufteilung des Images in die Partitionen und Untersuchung der freien Bereiche . . . 7
  - III-B5 Datenwiederherstellung mit foremost und unrm . . . 7
  - III-B6 Strings . . . 7
  - III-B7 Überprüfen der Signaturen . . . 7
- III-C Datenauswertung . . . 7
  - III-C1 Vorhandene Daten im Home Verzeichnis . . . 7
  - III-C2 zeitlicher Ablauf . . . 8
- III-D Analyse OS / Umgebung . . . 10
  - III-D1 Angriffe auf das System . . . 10

IV Anhang 11
- IV-A Logs . . . 11
  - IV-A1 auth_log . . . 11
  - IV-A2 dmesg . . . 27
- IV-B Dateien . . . 32
  - IV-B1 passwd . . . 32
  - IV-B2 bash_history robert.de.rainault . . . 32
  - IV-B3 bash_history robert.hut . . . 33
- IV-C Dokumentation der Arbeit . . . 33
  - IV-C1 Überprüfen der Signaturen . . . 33
  - IV-C2 Ausschneiden der Partitionen mit dd . . . 36
  - IV-C3 Strings auf Partitionen . . . 38
  - IV-C4 md5-Summen bilden . . . 49
- IV-D Hexdump Auszüge . . . 51
  - IV-D1 hexer Partitionstabelle . . . 51
- IV-E Bilder . . . 51

Abbildung 7.3: Gliederung von Bericht 1.

Die Zusammenfassung (Abschnitt II des Berichts) enthält eine stichpunktartige Aufzählung der auf dem untersuchten System vorgefundenen Benutzer.

II-A. Stichpunktartige Analyse des Endzustands

Es wurde das Linux-Betriebssystem Ubuntu 8.04 Hardy gefunden. Das System war so konfiguriert, dass es von außen ferngewartet werden konnte (über ssh). Das System wurde in einer virtuellen Umgebung genutzt.

Folgende Benutzerkonten wurden vorgefunden:

– defauft ID: 1000

– poor ID: 1004

– rich ID: 1003

– robert.de.rainauft ID: 1002

– robert.hut ID: 1001 defauft

- defauft

  – Das Benutzerkonto verfügt über administrative Rechte.

  – Der Benutzer hat sich mehrfach von außen eingeloggt und dabei die Administratorrechte benutzt.

  – Er hat die anderen Benutzer angelegt.

  – Er hat das Hauptverzeichnis von robert.hut gelöscht.

  – Er hat den Rechner heruntergefahren.

- poor

  – Dem Benutzer konnte kein Loginversuch nachgewiesen werden.

  – Es wurden mehrere Dateien im Hauptverzeichnis des Benutzers vorgefunden.

  * Cheque-Dateien mit zugehöriger Signatur.

  * Die Auswertung der Signaturen ergab keine Auffälligkeiten. Details: [. . .]

- rich

  – Dem Benutzer konnte kein Loginversuch nachgewiesen werden.

  – Es befanden sich am Ende keine Daten mehr in dem Hauptverzeichnis.

  – Allerdings wurden Spuren von gelöschten Dateien gefunden.

- robert.de.rainauft

  – Der Benutzer hat sich mehrfach von außen eingeloggt. [. . .]

  – Weiterhin ließ sich hier ein Ordner mit öffentlichem und privatem Schlüssel von robert.de.rainauft finden. Diese wurde zum Signieren der Dateien benutzt. Dies war für alle Benutzer möglich, da die Leserechte auf public gesetzt wurden.

- robert.hut

  – Der Benutzer hat sich mehrfach von außen eingeloggt.

  – Datei treasury mit Signaturdatei. Signatur stimmt nicht überein. Details [. . .]

  – Das Hauptverzeichnis des Benutzers wurde vom Benutzer defauft gelöscht.

  – Benutzer existiert aber auf dem System mit Logindaten.

Anschließend wird ein Überblick über die zeitliche Einordnung der Abläufe gegeben.

II-B. Zeitlicher Ablauf

An dieser Stelle werden die wichtigsten Aktionen zusammengefasst, für eine genaue Analyse der Dateiaktionen siehe III-C. Die Analyse zeigte, dass die beiden auf dem System vorhandenen Benutzer rich und poor eher eine passive Rolle gespielt haben. Nur den Benutzern robert.hut und robert.de.rainauft ließen sich Aktionen auf Userdateien zuordnen. Dann hat robert.de.rainauft zuerst cheques mit den Nummern 6–9 mit gültiger Signatur ausgestellt und sie dem Benutzer rich zukommen lassen. Danach

hat robert.hut die Dateien, die nötig sind, um eine gültige Signatur des Benutzers robert.de.rainauft zu erstellen, aus dessen Hauptverzeichnis zu sich kopiert und damit weitere gültige cheques mit den Nummern 0–4 ausgestellt. Sowohl die cheques aus dem Verzeichnis von rich also auch die neu erstellten wurden dann vermutlich von robert.hut in das Hauptverzeichnis des Benutzers poor kopiert. Für eine genauere Analyse der Zuordnung der Aktionen zu robert.hut siehe III-C2. Weiterhin hat robert.de.rainauft eine Datei treasury angelegt und ebenfalls signiert. Wiederum ist davon auszugehen, dass robert.hut die Datei verändert hat, allerdings ohne die zugehörige Signatur anzupassen. Das Nicht-Übereinstimmen der Signatur weist explizit darauf hin, dass die Datei in der vorliegenden Version nicht von dem Eigentümer der Signatur erstellt wurde. Die Analyse zeigt, dass vermutlich robert.hut die Datei mit einem neuen Wert überschrieben hat.

7.5.2 Bericht 2

Abbildung 7.4 auf der nächsten Seite und Abbildung 7.5 auf Seite → zeigen das Inhaltsverzeichnis von Bericht 2. Dieser Bericht zeichnete sich vor allem durch einen gut strukturierten Anhang aus.

7.5.3 Bericht 3

Es folgt ein Auszug aus Bericht 3. Dieser Bericht weist leider Unstimmigkeiten in der ***chain of custody*** auf. Außerdem wird nicht genau zwischen „Image“ und „CD“ unterschieden.

Chain of Custody

Nachdem mir das Image, in Form einer CD, von Frau B. am 16.03.2010 übergeben wurde, habe ich es in meinem Rucksack sicher in meine Wohnung transportiert. In meiner Wohnung legte ich das Image in einen nur mir zugänglichen und verschlossenen Schrank. Am 08.03.2010 nahm ich das Image wieder aus dem Schrank, legte es in meinen Rucksack und transportierte es ins forensische Labor. Dort legte ich die CD in die forensische Workstation in Raum 123. Der Raum ist stets verschlossen und nur Arbeitern und anderen Forensikern zugänglich. Nach vollendeten

forensischen Arbeiten transportierte ich die CD in meinem Rucksack wieder in meine Wohnung, wo sie in den verschließbaren, nur mir zugänglichen Schrank eingeschlossen wurde.

Inhaltsverzeichnis

I. Untersuchungsbericht **4**

1. Prolog **5**
1.1. Beweismittel 5
1.1.1. Identifikation 5
1.1.2. Verlauf 5
1.2. Auftrag 6
1.3. Arbeitsumgebung 6

2. Ergebniszusammenfassung **7**

3. Detailierter Ermittlungsverlauf **8**
3.1. Zusammenfassung 8
3.2. Erzeugung einer Arbeitskopie 9
3.3. Auswertung des Master Boot Records 9
3.4. Auswertung der Partitionstabelle 9
3.5. Erzeugung von Partitionskopien 10
3.6. Untersuchung der Partitionskopien 10
3.6.1. Untersuchung der ersten Partition 10
3.6.2. Untersuchung der zweiten Partition 12
3.6.3. Untersuchung der dritten Partition 13

II. Anhang **16**

4. Beweismittel **17**
4.1. Bild des Beweismittels 17

5. Arbeitsumgebung **18**
5.1. Verwendete Werkzeuge 18

Abbildung 7.4: Inhaltsverzeichnis von Bericht 2 (erster Teil).

6. Konsolenausgaben **19**
6.1. Master Boot Record . . . 19
6.2. Erste Partition . . . 19
6.2.1. Bootsektor . . . 19
6.2.2. Master File Table . . . 20
6.2.3. Gelöschte Dateien . . . 21
6.2.4. Dateiinhalte . . . 21
6.2.5. File carving . . . 25
6.3. Zweite Partition . . . 25
6.3.1. Bootsektor . . . 25
6.3.2. Master File Table . . . 26
6.3.3. Gelöschte Dateien . . . 27
6.3.4. File carving . . . 27
6.4. Dritte Partition . . . 27
6.4.1. Bootsektor . . . 27
6.4.2. Dateiinhalte ext2 Partition . . . 28
6.4.3. Dateiinhalte minix Partition . . . 28
6.4.4. File carving . . . 30
6.5. Log-Datei . . . 31

Abbildung 7.5: Inhaltsverzeichnis von Bericht 2 (zweiter Teil).

7.5.4 Bericht 4

Es folgt ein Auszug aus Bericht 4. Dieser Bericht enthält wertende Beurteilungen („gravierend“) und gleicht am Ende einem Plädoyer.

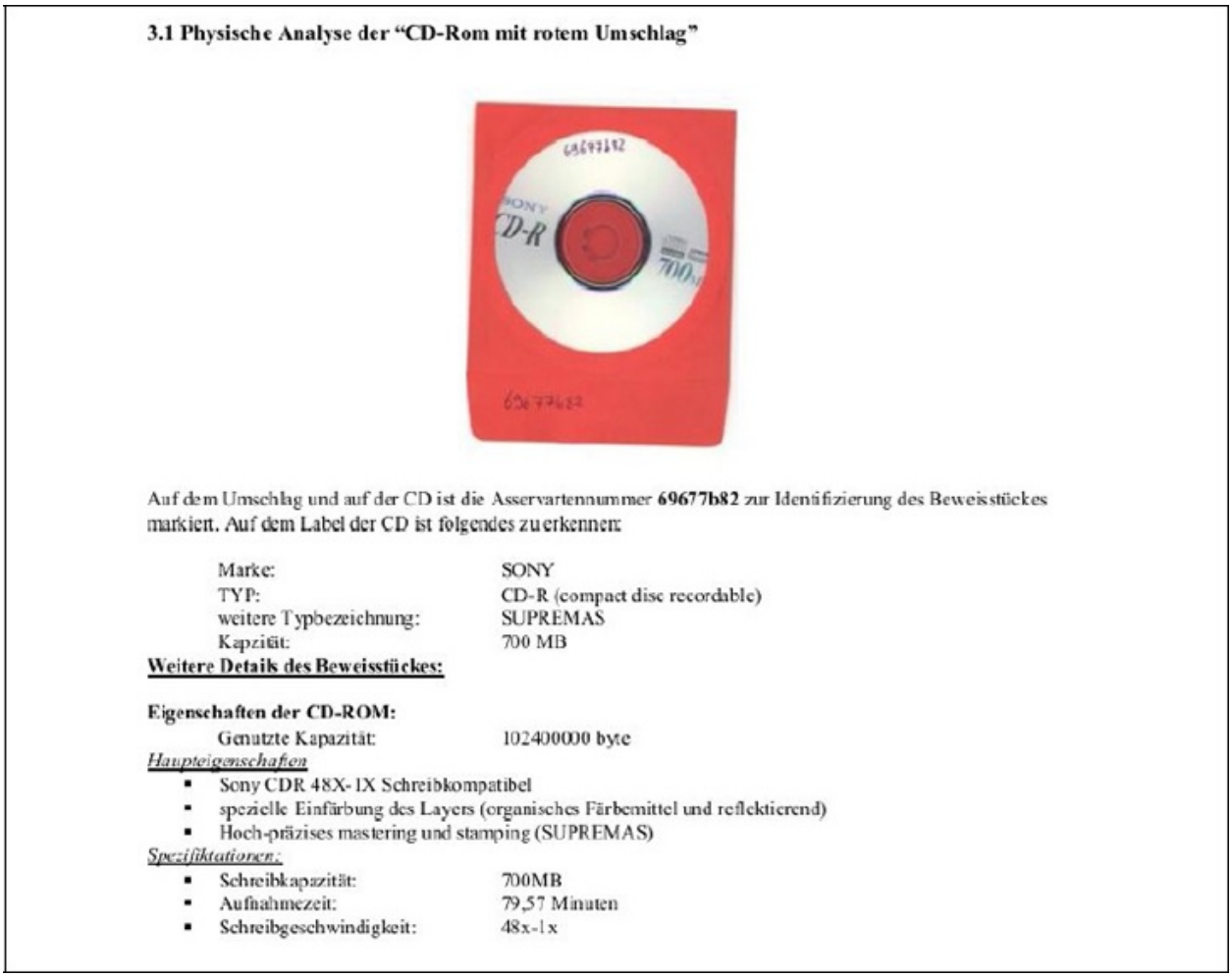
> Auf der Festplatte konnten acht Dateien mittels foremost wiederhergestellt werden. Es handelt sich um jpg- und pdf-Dateien. Die Bilddateien zeigen zweimal Dagobert Duck sowie zwei zubereitete Speisen und ein Bild eines winkenden Mannes (siehe 4.1 Bilder). Gravierender ist die Datei 12345.jpg, welche eine schematische Darstellung einer Bombe beinhaltet.
>
> [. . .]
>
> Zusammenfassend lässt sich sagen, dass die gefundenen Texte von Täterwissen zeugen, welches kein anderer in dieser Detaillierung haben könnte. In Kombination mit der Liste an Bauteilen für eine Zündungsvorrichtung und der schematischen Darstellung einer Bombe sowie der minutengenauen Angabe der Explosion lässt sich ein klarer Zusammenhang der Person XY mit den vorliegenden Erpressungen bestätigen.

7.5.5 Bericht 5

Ein abschließendes Beispiel ist in Abbildung 7.6 dargestellt. Der Ausschnitt des Berichtes zeigt eine Abbildung mit einer CD. Diese CD wird so dargestellt, als handele es sich um ein physisches Beweismittel. Allerdings handelt es sich lediglich um die CD-Rom, auf der die Festplattendaten (das ***image***) transportiert wurden, die Grundlage der Untersuchung waren. Das Aussehen der CD ist also für den Gang der Untersuchung nicht hilfreich. Der Transportweg ist eher Teil einer textuellen Beschreibung als Teil der Verwahrungskette.

7.6 Zusammenfassung

Dieses Kapitel betrachtete ausgewählte Aspekte der Dokumentation von digitalen Ermittlungen. Wir haben viele Kriterien kennen gelernt, die eine gute Dokumentation ausmachen. Diese Kriterien möchten wir hier in Form einer Checkliste nochmals zusammenfassen:



3.1 Physische Analyse der "CD-Rom mit rotem Umschlag"

Auf dem Umschlag und auf der CD ist die Asservatennummer **69677b82** zur Identifizierung des Beweisstückes markiert. Auf dem Label der CD ist folgendes zu erkennen:

Marke:	SONY
TYP:	CD-R (compact disc recordable)
weitere Typbezeichnung:	SUPREMAS
Kapzität:	700 MB

Weitere Details des Beweisstückes:

Eigenschaften der CD-ROM:

Genutzte Kapazität:	102400000 byte

Haupteigenschaften

- Sony CDR 48X-1X Schreibkompatibel
- spezielle Einfärbung des Layers (organisches Färbemittel und reflektierend)
- Hoch-präzises mastering und stamping (SUPREMAS)

Spezifiktationen:

Schreibkapazität:	700MB
Aufnahmezeit:	79,57 Minuten
Schreibgeschwindigkeit:	48x-1x

Abbildung 7.6: Auszug aus Bericht 5.

1. Anforderungen an die äußere Form:

    - Aussagekräftige Kopf- und Fußzeile
    - Seitenzahlen, übersichtlicher Seitenrand und Zeilenabstand
    - Gliederung mit referenzierbarer Nummerierung
    - Bei umfangreicheren Berichten: Inhalts-, Abbildungs-, Tabellenverzeichnisse hinzufügen

- Wenn nötig, sollten auch Literaturhinweise und Quellennachweise angeben werden

2. Anforderungen an den Inhalt:

    - Berichtskopf/Titelseite mit Angaben zu

        – Art des Berichtes (Gutachten, Untersuchung, Protokoll, etc.)

        – Version, Bearbeitungsstand, Datum

        – Angaben zum Autor bzw. Auftragnehmer (Autoren, Organisationseinheit, Dienststelle)

        – Kontext des Berichts, wie zum Beispiel Aktenzeichen oder Verfahrensnummern

        – Versions- bzw. Änderungshistorie

    - Prolog

        – Angabe zum Auftraggeber, Datum der Beauftragung

        – Untersuchungsauftrag und Untersuchungskontext

        – Auflistung der untersuchten Spuren mit ihren jeweiligen Identifizierungsmerkmalen

        – Beschreibung der Verwahrungskette (***chain of custody***)

        – Beschreibung der eigenen Arbeitsumgebung (Hardware, Software, insbesondere Softwareversionen)

    - Zusammenfassung für Nicht-Techniker/Managementzusammenfassung (executive ***summary***)

        – ***Umfang maximal eine Seite DIN A4***

        – Wesentliche Ergebnisse und Schlussfolgerungen mit Bezug zum Untersuchungsauftrag zusammenfassen

        – So wenig technische Details wie nötig, allgemeinverständliche Sprache, Vermeiden technischer Fachbegriffe

    - Zusammenfassung für Techniker

        – Umfang abhängig von der Untersuchung, allerdings sollten sich auch komplexere Untersuchungen meist auf maximal 10 Seiten

DIN A4 zusammenfassen lassen

– Wesentliche Untersuchungsschritte und Ergebnisse dokumentieren

– Logische Folge der Untersuchungsschritte begründen

– Wesentliche Untersuchungsschritte durch entsprechende Dokumentuntergliederung (Abschnitte) sichtbar machen

– Präzise Fachsprache verwenden

– Belege/Fundstellen möglichst genau angeben, um eine schnelle Überprüfbarkeit zu ermöglichen

- Falls nötig: Details für Techniker

  – ***Alle*** Details, entweder als eigenes Kapitel oder als separate Anlage, gegebenenfalls auch nur elektronisch auf CD beilegen

  – In aller Ausführlichkeit mit präziser Fachsprache

  – Längere Listings, Logdateien, Bildschirmfotos

  – Jeweils mit Nummern versehen, so dass sie aus dem Hauptteil des Berichts referenzierbar sind

3. Weitere Anforderungen

   - Korrekter Ausdruck, korrekte Rechtschreibung und Grammatik
   - Präzise aber doch einfache und verständliche Sprache
   - Wissenschaftliche Standards beim Zitieren anwenden
   - Nicht nur beschreiben, was man getan hat, sondern auch warum
   - Wiederholbarkeit und Überprüfbarkeit ermöglichen
   - Vermutungen kennzeichnen, juristischen Einschätzungen vermeiden

Diese Hinweise müssen natürlich auf den Einzelfall angepasst werden, können aber als knappe Orientierung dienen. In der Praxis muss man selbstverständlich einen Kompromiss aus Umfang und Aussagekraft der Dokumentation finden. Zudem haben sich in vielen Bereichen der Praxis sowieso eigene Standards für die Berichterstellung etabliert. Zusammenfassend kann das Motto bei

forensischen Untersuchungen aber nur lauten: „Dokumentieren, dokumentieren und nochmals dokumentieren!“

Kapitel 8

Praktische Aspekte digitaler Ermittlungen

Autoren: Felix Freiling, Andreas Dewald

Abgesehen vom groben, einheitlichen Rahmen eines Vorgehensmodells, ist jede digitale Ermittlung einzigartig. Deswegen müssen bei jeder Untersuchung technische und kriminalistische Expertise und Erfahrung zusammenwirken. Dieses Kapitel gibt einen Überblick über verschiedene praktisch bedeutsame Aspekte digitaler Ermittlung. Es geht hierbei einerseits um organisatorische Aspekte, also etwa um die Rollen und Aufgabenverteilungen im Rahmen einer Ermittlungen. Andererseits behandeln wir das praktisch bedeutsame Problem des Umgangs mit immer weiter zunehmenden Datenmengen im Rahmen einer solchen Ermittlung.

8.1 Organisatorische Aspekte

Die folgenden Erkenntnisse sind das Ergebnis strukturierter Interviews mit Praktikern aus dem Bereich der Strafverfolgung in Deutschland. Die Übertragbarkeit auf (nicht-öffentliche) Ermittlungen innerhalb von Organisationen, also etwa interne Ermittlungen innerhalb von Firmen, ist daher nicht unbedingt gegeben.

8.1.1 Rollen und Aufgabenverteilung

Im Bereich der Strafverfolgung gibt es verschiedene Rollen im Rahmen einer digitalen Ermittlung. Wie bei einer klassischen (nicht-digitalen) Straftat wird die eigentliche Ermittlung von einem Kriminalkommissar oder Staatsanwalt geführt. Diese sind die ***Ermittler*** im eigentlichen Sinne. Die Ermittler führen die Beschlagnahme potentieller Spuren durch, die im Sprachgebrauch der

Strafverfolgung als ***Beweismittel*** bezeichnet werden. Die Beweismittel werden zur kriminaltechnischen Untersuchung in ein entsprechendes Labor gegeben. Hierzu ist meist ein Auswerteantrag erforderlich, in dem festgelegt wird, woraufhin das Beweismittel untersucht werden soll. Im Labor analysieren Spezialisten das Beweismittel und fertigen einen Bericht über die Ergebnisse dieser Untersuchung für den Ermittler an. Der Ermittler kann die Ergebnisse der kriminaltechnischen Untersuchung nun für seine Ermittlungen nutzen.

In Bezug auf diese (allgemeine) Vorgehensweise stellt aktuell auch die forensische Informatik keine Ausnahme dar: Datenträger oder andere Geräte werden meist wie gewöhnlich bei einer Durchsuchung von den Ermittlern sichergestellt. Die Auswertung im Labor geschieht dann durch forensische Informatiker. Nur wenn es triftige Gründe für die Annahme gibt, dass die Unterstützung von Spezialisten bereits bei der Sicherstellung der Geräte erforderlich ist, werden die Ermittler schon bei der Beschlagnahme von der entsprechenden Serviceeinheit der Polizei unterstützt.

Der Ermittler stellt dann einen Antrag auf Auswertung durch die Spezialisten mit einer Beschreibung der gesuchten digitalen Spuren. Meist umfasst dies lediglich gesuchte Dateitypen und eventuell eine Stichwortliste. Über den konkreten Sachverhalt weiß der forensische Informatiker überlicherweise nur sehr wenig. Die gefundenen digitalen Spuren werden zusammen mit einem Bericht wiederum dem Ermittler zur Verfügung gestellt, der diese Daten dann sichten und für die weitere Ermittlungsarbeit nutzen kann.

Diese Aufgabenverteilung bietet den Vorteil, dass die Spezialisten der kriminaltechnischen Untersuchung nicht in die Ermittlung involviert sind und dadurch leichter einen objektiven Bericht verfassen können. Meist ist diese Aufteilung in ermittlungsführende Dienststelle und Serviceeinheit auch erforderlich, da großer Bedarf an digital-forensischen Untersuchungen besteht, aber in der Regel weit mehr Sachermittler als forensische Informatiker zur Verfügung stehen. Um die Spezialisten in den gesamten Ermittlungsprozess einbeziehen zu können, wären also eine weit größere Zahl von Forensikern erforderlich. Neben den erhöhten Personalkosten steht dem aktuell vor allem ein Mangel an entsprechend ausgebildeten Spezialisten im Wege.

Ein Nachteil dieser Aufgabenverteilung ist, dass die Techniker nicht wissen, wonach die Ermittler ***inhaltlich*** suchen. Sie können lediglich die im Auswerteantrag angeführten Daten erheben und zur Verfügung stellen. Die Ermittler verfügen aber oft nicht über hinreichendes Verständnis von digitalen Systemen, um einschätzen zu können, wo und in welcher Form sich die

gesuchten digitalen Spuren auf den sichergestellten physischen Beweismitteln befinden könnten. Hierdurch kann es vorkommen, dass digitale Spuren nicht gefunden werden, weil sie aus dem vom Ermittler festgelegten Raster fallen. Hier wird auch direkt der Vorteil einer engeren Verzahnung von Sicherung, Erhebung und Aufbereitung digitaler Spuren und der Ermittlung deutlich: Die Möglichkeit einer gezielteren Auswertung der sichergestellten Geräte. Hilfreich ist daher ein Grundverständnis für die forensische Informatik bei allen Ermittlern, um konkretere Anforderungen an die Serviceeinheit formulieren zu können. Beispiele für solche Fragen haben wir in Abschnitt 2.5.3 auf Seite → in Kapitel 2 bereits behandelt.

Ein Vorteil der Kooperation von Ermittler und Techniker bereits bei der Beschlagnahme ist jedoch die Möglichkeit der Hilfestellung durch den Experten bei der Entscheidung über die Art und Weise der Beschlagnahme. Der forensische Informatiker kann eine fachliche Einschätzung der Chancen und Risiken einer körperlichen Beschlagnahme bzw. einer logischen Sicherung vor Ort abgeben. Die Risiken beziehen sich im Wesentlichen auf einen möglichen Datenverlust oder eine unbeabsichtigte Manipulation digitaler Spuren.

8.1.2 Komplexe Durchsuchungen

Bei Durchsuchungsmaßnahmen in Unternehmen oder Rechenzentren, bei denen IT in großem Ausmaß vorgefunden wird, findet eine sogenannte ***Triage*** statt. Diese Triage wird meist von der Serviceeinheit für IT-Beweismittelsicherung durchgeführt. In den meisten Fällen tritt die Firma selbst als Zeuge auf, so dass bei der Durchsuchung mit dem Unternehmen kooperiert werden kann. Ist dies nicht der Fall, tritt direkt ein hinreichend großes Team zur Beweismittelsicherung mit einem Durchsuchungsbeschluss auf und versucht zunächst, den Zustand der vorgefundenen Systeme zu erhalten. Insbesondere muss dafür gesorgt werden, dass kein Mitarbeiter des Unternehmens mehr Einfluss auf die betreffenden Systeme nehmen kann. Dies ist jedoch nur selten erforderlich, so dass meist zunächst der Vorstand vor Ort informiert und ein „runder Tisch“ gebildet wird. An diesem runden Tisch beraten dann üblicherweise der Vorstand, die Rechtsabteilung und Administratoren mit den Ermittlern über erforderliche Maßnahmen und deren Durchführung. Die ständige Kommunikation auch während der Durchführung der Spurensicherung spielt hier eine besonders große Rolle.

Bevor das Team zur IT-Beweismittelsicherung mit der eigentlichen Sicherung beginnen kann, muss zunächst geklärt werden, wer die Daten sichert und welche Daten es überhaupt gibt. Anschließend muss über die konkrete Sicherungsreihenfolge und die Sicherungstechnik entschieden werden.

Die Antwort auf die Frage, wer Daten sichern soll, hängt maßgeblich davon ab, ob das Unternehmen als Zeuge auftritt und kooperiert oder nicht. Ist keine Kooperation möglich, muss die Sicherung in jedem Fall durch das Team für die Beweismittelsicherung selbst oder durch damit beauftragte externe Dienstleister vorgenommen werden, falls die eigenen Kapazitäten nicht ausreichen. Meist kann jedoch auf die Unterstützung der firmeneigenen IT-Dienstleister zurückgegriffen werden. Dies ist auch für die Ermittler von Vorteil, da gerade die Sichtung gewachsener Systemlandschaften eine große Herausforderung darstellt. Gegebenenfalls kann auch bereits die telefonische Unterstützung durch den IT-Dienstleister des Unternehmens hilfreich sein.

Um sich einen Überblick über die vorliegenden Daten zu verschaffen, muss zunächst ermittelt werden, wo Daten physisch gespeichert sind (beispielsweise auf Servern, Bandautomaten oder Ähnlichem) und wie der Zugriff auf diese Daten erfolgen kann. Hierzu wird meist ein Netzwerkplan über die gesamte Systemlandschaft angefertigt. Dabei kann es durchaus notwendig sein, der Verkabelung im Haus zu folgen, um diejenigen Geräte vollständig zu erfassen, die über ein kabelgebundenes Netzwerk miteinander verbunden sind.

Das Auffinden drahtlos angebundener Geräte hingegen gestaltet sich oft schwieriger. Eine Möglichkeit zur Enumeration von WLAN-Geräten stellt die Auswertung der DHCP-Client-Liste oder der Log-Dateien des WLAN-Routers oder Access Points dar. In besonders schweren Fällen, wie beispielsweise organisierter Kriminalität, besteht mit entsprechenden zusätzlichen richterlichen Beschlüssen auch die Option, Netzwerkverkehr mitzuschneiden (***sniffing***), um auf aktive WLAN-Geräte aufmerksam zu werden und diese anhand der Signalstärke aufzufinden. Natürlich muss auf die Verhältnismäßigkeit der zu ergreifenden Maßnahmen geachtet werden. Glücklicherweise sind aber insbesondere Netzwerkspeicher (beispielsweise Storage Area Networks, File Server) aus Performance-Gründen meist über Kabel an das Netzwerk angebunden. Das Horror-szenario einer eingemauerten WLAN-Netzwerkfestplatte ist also in der Praxis selten anzutreffen.

Eine ähnlich große Herausforderung stellt das Auffinden von kleinen Speichermedien, wie beispielsweise USB-Sticks oder Speicherkarten, dar. Es gibt sie in jeder nur erdenklichen Form: Ob Autoschlüssel, Taschenmesser,

Schmuck oder Schreibutensil, nahezu jeder noch so kleine Gebrauchsgegenstand kann zur Tarnung für einen Speicher-Stick werden. Doch damit nicht genug: Durch ihre geringe Größe lassen sich kleine Speichermedien überall verstecken. Regelmäßig werden sie auf Lampen oder mit einem Kaugummi unter einem Möbel befestigt sichergestellt. Die extrem flachen Micro-SD-Karten beispielsweise können sogar zwischen den Seiten eines Buches versteckt werden. Ermittler müssen also ein gutes Gespür für mögliche Verstecke und Tarnungen dieser Datenträger entwickeln. Bei entsprechend schweren Fällen bleibt bei einer Durchsuchung lediglich die Möglichkeit, wirklich jeden Gegenstand und jedes potentielle Versteck zu untersuchen.

Bei der Auflistung der Beweismittel dürfen natürlich auch andere Geräte als nur Computer und Speichermedien nicht vergessen werden. Es existiert bereits eine Vielzahl an Geräten, die in der Lage sind, relevante Daten zu speichern, obgleich ein Ermittler dies nicht unmittelbar vermuten würde. Immer häufiger müssen auch Drucker, digitale Kopierer und Telefonanlagen ausgewertet werden. Diese Liste ist bei Weitem nicht vollständig und wächst zudem beständig. Schon heute gehört das Auswerten von Mobiltelefonen, MP3-Playern und sogar Spielekonsolen zum Alltag der Ermittler. In Zukunft könnten auch die Kaffeemaschine, der Kühlschrank und der Fernseher (Smart-TVs) in den Fokus einer digitalen Ermittlung gelangen.

Ein weiteres Problem und eine rechtliche Grauzone stellen Onlinespeicher dar. Stellt der Ermittler fest, dass auf einem laufenden System Onlinespeicher eingebunden sind, kann nach Rücksprache mit dem Staatsanwalt zunächst der Inhalt des Speichers vor Ort logisch gesichert werden. Wie bei allen Eingriffen in ein laufendes System, gilt es, besonders auf eine genaue Dokumentation des Vorgehens zu achten, um die Verwertbarkeit der gesicherten digitalen Beweismittel vor Gericht zu gewährleisten. Die rechtliche Frage, ob diese Daten aufgrund des vorliegenden Durchsuchungsbeschlusses trotz der räumlichen Entfernung sichergestellt werden dürfen, kann nur ein Staatsanwalt entscheiden. Die aktuell herrschende Meinung ist, dass logisch gesichert werden darf, worauf von einem vor Ort befindlichen Computer direkter logischer Zugriff besteht. Dieses Thema wird aber sicherlich in der nahen Zukunft immer wichtiger werden, da es bereits einige Zeit immer stärke Tendenzen dazu gibt, Server im Ausland zu mieten. Das Thema Cloud-Forensik möchten wir an dieser Stelle bewusst nicht aufgreifen.

8.2 Priorisierung und Auswahl bei der Sicherung

Nachdem das Ermittlerteam sich eine möglichst vollständige Übersicht über die vorhandenen Daten verschafft hat, schließt sich die Frage an, welche der Daten gesichert werden sollen. Diese Entscheidung wird maßgeblich durch eine Art Lastenheft des Sachermittlers beeinflusst, in dem festgehalten wird, welche Daten von Interesse sind. Vor Ort muss dann im Einzelfall konkret entschieden werden, ob ein bestimmtes Beweismittel gesichert werden soll oder nicht. In beinahe allen strafrechtlichen Fällen ist hierzu ein Staatsanwalt anwesend oder zumindest telefonisch mit dem Team in Kontakt. Dieser ist in solchen Fällen auch der „Auftraggeber“ und entscheidet letztlich darüber, welche Daten sichergestellt werden.

Allerdings sind bei der Auswahl und der anschließenden Priorisierung bei der Verarbeitung viele technische und methodische Rahmenbedingungen zu beachten, auf die in diesem Abschnitt eingegangen werden soll. So stößt die Verarbeitung der zunehmenden Datenmengen mittlerweile an technische Grenzen. Beispielsweise dauert das bitweise Sichern einer einzelnen 3-Terabyte-Festplatte mit aktuellen Technologien (Schreibblocker mit maximaler USB3.0 Bandbreite) etwa 1,5 Stunden. In Unternehmen mit hunderten Terabyte an Datenspeicher wird dies zum Problem. Außerdem muss auf den Grundsatz der Verhältnismäßigkeit geachtet werden. Ist das gesamte Unternehmen „auf der Anklagebank“, kann auch eine vollständige körperliche Beschlagnahme der Infrastruktur in Betracht gezogen werden. Tritt das Unternehmen lediglich als Zeuge auf oder wäre der Aufwand für eine körperliche Beschlagnahme zu groß, kann die logische Sicherung eines Teils der Daten ausreichen.

Nach der Diskussion von grundsätzlichen Erwägungen werden wir auf verschiedene technische und organisatorische Maßnahmen eingehen, die bei der Sicherung digitaler Spuren helfen sollen. Die Ausführungen beziehen sich in wesentlichen Teilen nur auf die Sicherung von persistenten Speichermedien wie Festplatten und USB-Sticks.

8.2.1 Grundsätzliche Erwägungen

Wie bereits vielfach angemerkt, muss bei der Beweismittelsicherung und -analyse großer Wert darauf gelegt werden, dass Spuren nicht verändert werden. Im Gegensatz zu physischen Spuren kann man aber von digitalen Spuren inhaltlich identische Kopien erstellen, ohne das Original zu verändern. Es ist

also für die Ergebnisse der Untersuchung gleichgültig, ob man auf dem Original oder der Kopie arbeitet. Um sich nicht der Gefahr einer Veränderung auszusetzen, arbeiten Ermittler bei der Analyse daher oft mit der Kopie statt mit dem Original. Wie in Kapitel 6 angemerkt, gehört dieses Vorgehen zum Standard in der forensischen Informatik, ist aber aus Verhältnismäßigkeitsgründen vermehrt unter Beschuss.

In der Vergangenheit wurde standardmäßig die Technik des Spiegelns angewendet (***imaging***). Dazu wird der komplette Inhalt eines Speichermediums 1:1 (also Bit für Bit) auf ein anderes Speichermedium (meist in eine Datei) geschrieben. In der Praxis stößt die Technik des Spiegelns inzwischen an Grenzen. Ermittler sind zunehmend mit großen und komplexen Datenmengen konfrontiert, deren Bearbeitung neben einem wachsenden Zeitaufwand bei der Analyse auch eine zunehmende Menge an Speicherressourcen zur Archivierung erfordert. Bereits für die eigentliche Sicherung von Datenträgern (insbesondere von Festplatten) wird mit zunehmender Speicherkapazität auch ein längerer Zeitraum beansprucht. Die Kosten einer Analyse steigen jedoch überproportional mit der Datenmenge. Auch aus Gründen der Verhältnismäßigkeit des Eingriffes in die Privatsphäre des Beschuldigten gilt, dass Daten nur im minimal notwendigen Umfang gesichert werden sollten. Es ist also in der Praxis mehr als bisher darauf zu achten, bei der Auswahl der zu sichernden Daten und der späteren Sicherung ein flexibles Instrumentarium einzusetzen, wie es etwa im Kontext der ***selektiven Sicherung*** in Kapitel 6 beschrieben und hier kurz wiederholt wird.

8.2.2 Technische Rahmenbedingungen

Der Speicherplatz auf Speichermedien ist in der Regel in Partitionen aufgeteilt, in denen jeweils Dateisysteme liegen. Innerhalb dieser Dateisysteme gibt es dann Dateien und Verzeichnisse. Die Daten auf einem Speichermedium liegen also in „vertikal eingebetteten“ Schichten oder Ebenen (siehe Abbildung 8.1). Da häufig jede Partition auch ein Dateisystem enthält, unterscheiden wir in der folgenden Diskussion der Einfachheit halber statt vier nur drei Ebenen, und zwar:

1. die ***physische Ebene***, also die Bits auf der „nackten Festplatte“,
2. die ***Partitionsebene***, also die Teilmenge an Bits der physischen Schicht, welche sich innerhalb einer Partition befinden, und

3. die ***Dateiebene***, also die Menge der Bits innerhalb einer Partition, welche durch das Dateisystem als Dateien oder Verzeichnisse organisiert sind.

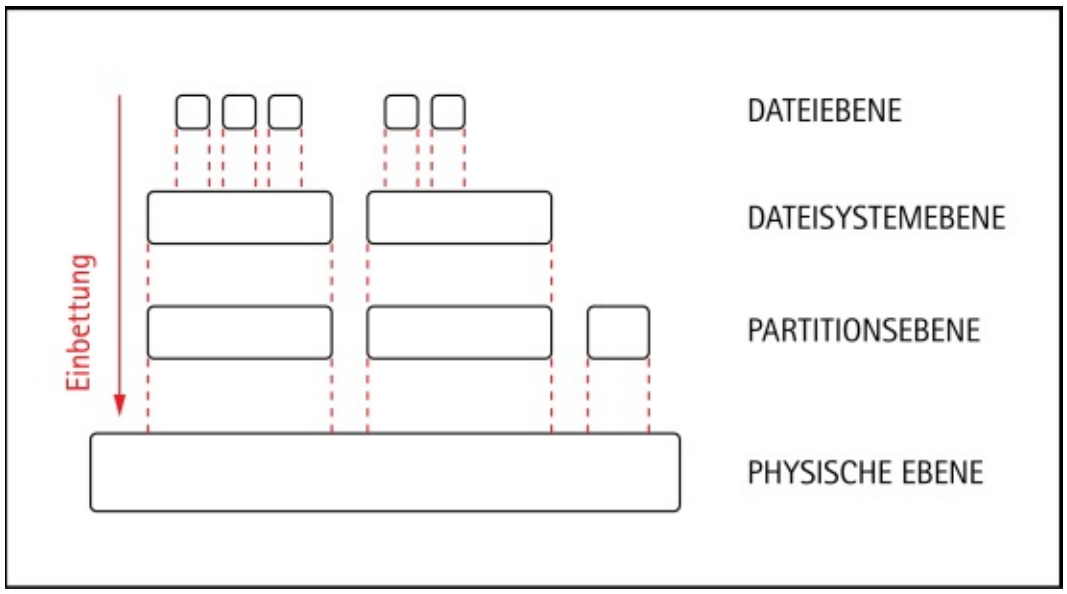


Abbildung 8.1: Hierarchische Gliederung von Speichermedien.

Entsprechend dieser Gliederung gibt es verschiedene Ebenen, auf denen eine Auswahl und eine Priorisierung der zu sichernden Daten vorgenommen werden kann. Diese Ebenen legen verschiedene technische Methoden nahe, die sich zwischen zwei Extrempositionen bewegen: Auf der einen Seite steht das „Maximum“ an Daten, das zu sichern wäre (also jedes verfügbare Bit). Dies bedeutet eine Sicherung auf der physischen Ebene. Auf der anderen Seite steht das „Minimum“ an Daten, also ausschließlich die für die Ermittlung relevanten Bits. Welche Bits für die Ermittlung relevant sind, ist stark vom Einzelfall abhängig und methodisch auch noch nicht richtig erforscht.

8.2.3 Aktuelle Praxis bei der Datensicherung

In der Praxis der Strafverfolgungsbehörden dominiert derzeit die Extremposition des „Maximums“, also die Sicherung von Speichermedien auf der physischen Ebene. In der Regel wird jedoch eine Vorselektion durchgeführt, indem einzelne Rechner oder Speichermedien vom Ermittler vor Ort als relevant oder irrelevant eingestuft werden. Als irrelevante und somit nicht zu sichernde Daten werden heute beispielsweise häufig Musik-CDs oder Installationsmedien eingestuft. Eine Ausnahme von dieser generellen Vorgehensweise stellen Sicherungen von

umfangreichen IT-Systemen wie beispielsweise Serverlandschaften in Rechenzentren dar. In solchen Fällen müssen in der Praxis Einschränkungen der zu sichernden Daten vorgenommen werden, sofern sich diese nicht bereits aus dem rechtlichen Rahmen ergeben. Das bedeutet, dass nicht mehr auf der physischen, sondern auf der Dateisystem- oder der Dateiebene gesichert wird.

In der Praxis privater Ermittlungen und der Bewältigung von Angriffen und Sicherheitsvorfällen haben sich in der Vergangenheit Techniken etabliert, die eine Komplettsicherung vermeiden. Die Komplettsicherung (also die Sicherung auf der physischen Ebene) wird hier nur als eine Möglichkeit unter vielen gesehen, die nur dann notwendig ist, wenn das Beweismittel in einem Straf- oder Zivilverfahren verwendet werden soll. Hier wirken die Standards der Strafverfolgung zurück auf die Standards der Privatwirtschaft.

8.2.4 Drei grundsätzliche Sicherungsmethoden im Überblick

Im Folgenden stellen wir eine Auswahl an Sicherungsmethoden vor, die sich zwischen den Extrempositionen „Maximum“ und „Minimum“ bewegen. Die einzelnen Methoden sind abnehmend nach der Menge der gesicherten Bits sortiert, sie sichern also immer weniger Daten. Dies geht mit einer Sicherung auf zunehmend höheren Abstraktionsebenen einher (siehe Abbildung 8.1). Wie weiter unten noch dargestellt werden wird, sollte in der Praxis immer das „mildeste“ Verfahren gewählt werden, also das Verfahren, das so wenige Daten wie möglich sichert.

8.2.5 Sicherung auf der physischen Ebene

Die ausgewählten Datenträger werden auf der ***physischen Ebene*** gesichert. Bei der Sicherung wird der zu sichernde Datenträger in der Regel direkt an ein eigenständiges Computersystem (meist eine forensische Workstation) angeschlossen. Festplatten müssen dazu in der Regel aus Computersystemen ausgebaut werden, um sie direkt an eine forensische Workstation anschließen zu können. Durch den Einsatz eines eigenständigen Computersystems wird das Booten vom zu sichernden Datenträger vermieden, ein Vorgang, der in der Regel mit Schreibzugriffen auf das Medium verbunden ist. Bei jeder Sicherung wird zusätzlich ein Schreibschutz eingesetzt, der entweder auf Ebene der

Software oder in Hardware realisiert ist (***hardware write blocker***) und eine Veränderung des Originaldatenträgers verhindern soll.

Für die Erstellung einer forensischen Kopie können verschiedene kommerzielle oder frei verfügbare Software-Werkzeuge zum Einsatz kommen. Gängige kommerzielle Werkzeuge sind zum Beispiel ***EnCase*** (vom Hersteller Guidance Software aus den USA), ***X-Ways*** (von X-Ways Software Technology, Köln), sowie der ***Tableau Imager*** (von Tableau, USA) oder der ***FTK Imager*** (von AccessData, USA). Teilweise kommen auch frei verfügbare Werkzeuge zum Einsatz, wie etwa das Programm dd, das in vielen Unixbasierten Betriebssystemen auf der Kommandozeile zur Verfügung steht.

8.2.6 Sicherung auf der Partitionsebene

Hier wird das Speichermedium nicht auf der physischen Ebene, sondern auf der ***Partitions-*** beziehungsweise auf der ***Dateisystemebene*** gesichert. Dies erfolgt in der Regel mit den gleichen Werkzeugen wie die Sicherung auf der physischen Ebene.

Bei der Sicherung auf Partitionsebene muss allerdings vorher eine Entscheidung darüber getroffen werden, welche Partitionen zu sichern sind. Dazu benötigt man Informationen darüber, welche Partitionen es auf dem Datenträger gibt. Vor dem Auslesen der eigentlichen Daten muss also zunächst die Partitionstabelle des Datenträgers inspiziert werden. Diese ist eine Art Inhaltsverzeichnis der Festplatte. Prinzipiell kann dies auch mittels der zuvor beschriebenen Werkzeuge erfolgen. Allerdings ist es auch möglich, das zu sichernde Computersystem von einer speziellen Live-CD/-DVD zu starten. Dabei kopiert sich das Betriebssystem in den Hauptspeicher, ohne Schreibzugriffe auf angeschlossene Datenträger zu verursachen. Der zu sichernde Datenträger muss anschließend im schreibgeschützten Modus in das System eingebunden werden (***read-only mounted device***), damit der Zustand des Originaldatenträgers gewahrt wird. Falls der Rechner mittels Live-CD/-DVD gestartet oder in einem laufenden Zustand vorgefunden wurde und über einen Netzwerkanschluss verfügt, können die Daten auch über das Netzwerk gesichert (kopiert) werden.

Ermittler bevorzugen für die beiden bisher beschriebenen Methoden heute jedoch in der Regel einen eigenen Sicherungsrechner, da insbesondere bei der

Verwendung eines ***hardware write blocker*** die Gefahr versehentlicher Modifikationen ausgeschlossen werden kann.

8.2.7 Sicherung auf der Dateiebene

Hierbei handelt es sich um eine Sicherung auf Ebene der ***Dateien im Dateisystem***. Das Vorgehen bei der Sicherung folgt dem Ablauf, der bei der Sicherung auf Partitionsebene beschrieben wurde (Verwendung einer Live-CD/-DVD). Allerdings werden – im Unterschied zur Sicherung einer ganzen Partition – in diesem Fall nur die logisch vorhandenen, also die noch nicht gelöschten Dateien gesichert. Diese Methode ist mit dem Anfertigen einer Sicherheitskopie (Backup) für ausgewählte Dateien des Datenträgers vergleichbar. Bei einem herkömmlichen Backup kommt es jedoch in der Regel zu Veränderungen der Metadaten von Dateien, etwa den Zugriffszeiten.

Für die Sicherung ausgewählter Dateien auf einem Datenträger ist eine inhaltliche Bewertung der Daten durch die Ermittler „vor Ort“ erforderlich. Hierfür müssen die Daten gesichtet werden. Methodisch wird analog zu den beiden vorgenannten Alternativen vorgegangen, um zu gewährleisten, dass das originale Beweismittel nicht verändert wird. Da diese Methode nur die logisch vorhandenen Dateien sichert, ist sie nur angebracht, wenn man die Existenz von Dateien nachweisen möchte. Sie ist nicht wirksam, wenn die relevanten Informationen bereits gelöscht sein könnten.

Viele kommerzielle Software-Werkzeuge, wie etwa EnCase, lesen das Dateisystem eines zu sichernden Datenträgers aus und erstellen eine Vorschau auf dessen Inhalte. Bei der Sichtung und Auswahl von Inhalten sind Sortier- und Filterfunktionen besonders hilfreich. Filterfunktionen ermöglichen das Ausblenden von Dateien in Abhängigkeit von ihren Metadaten, etwa der Dateigröße oder den Zugriffszeiten.

8.2.8 Sicherung als Teil der Live-Analyse

Unter Umständen kann es notwendig sein, sowohl den Rechner also auch das Betriebssystem des Beschuldigten zu verwenden, um die Sicherung auf Dateiebene durchzuführen, etwa als Teil einer Live-Analyse. Aus technischer Sicht ist die Live-Analyse mit Nachteilen verbunden, da die Integrität des Beweismittels nicht garantiert werden kann. Dennoch gibt es in manchen Fällen keinen anderen direkten Zugang zu den Daten. Hat etwa der Beschuldigte die

Festplatte seines Rechners mit hinreichend starken Methoden verschlüsselt, so können gespeicherte Daten in der Regel nur solange gesichtet und gespeichert werden, wie die Verschlüsselung im laufenden Betrieb aufgehoben ist. Auch kann ein Bedürfnis bestehen, Inhalte von über das Netz verbundenen Speichermedien wie E-Mail-Servern, Netzfestplatten oder Cloud-Speichern mittels eines Remote-Zugriffs vom Rechner des Beschuldigten auszulesen, damit der durch die Durchsuchung gewarnte Beschuldigte die Daten nicht im unmittelbaren Anschluss löschen und so Beweismittel vernichten kann.

8.2.9 Beweiswert und Verhältnismäßigkeit

Zunächst möchten wir nochmals erwähnen, dass es von höchster Wichtigkeit ist, den gesamten Vorgang der Sicherung gut zu dokumentieren. Besonders Fotos spielen hierbei eine wichtige Rolle. Großverfahren dauern üblicherweise mehrere Jahre. Daher stellt die Dokumentation eine wichtige Gedächtnisstütze für den Ermittler dar, der letztlich vor Gericht als Gutachter auftritt.

In der Praxis dominiert zur Zeit in kleinen Verfahren die Sicherung auf physischer Ebene und in Großverfahren (Durchsuchung in Rechenzentren) die Sicherung auf Dateiebene. Aus rechtlicher Sicht haben die Daten aus allen genannten Alternativen prinzipiell denselben Beweiswert, gilt doch für die Bewertung der gesicherten Daten im Strafprozess der Grundsatz freier richterlicher Beweiswürdigung, die allerdings auf einer objektiven Grundlage beruhen muss. Dabei ist zu beachten, dass nicht nur die gesicherten Daten Beweismittel im Strafprozess sind. Hinzu tritt immer das Zeugnis der Ermittlungspersonen, die Auskunft über die Umstände geben können, unter denen diese Daten aufgefunden und gesichert wurden. Der die Vertrauenswürdigkeit limitierende Faktor bleibt immer auch der Zeuge und nicht ausschließlich die Technik (Bäcker u.a., 2010).

Die partielle oder vollständige Sicherung von Daten aus Speichermedien beeinträchtigt die Vertraufichkeit des betroffenen informationstechnischen Systems und greift daher stark in die Privatsphäre des Beschuldigten ein. Um die Verhältnismäßigkeit dieses Eingriffs zu wahren, muss er auf das erforderliche Maß begrenzt werden. Dies ist besonders dringlich, wenn Daten aus Systemen gesichert werden, die einer Vielzahl von Nutzern zur Verfügung stehen, von denen nur gegen einzelne ermittelt wird. Aber auch wenn die Durchsuchung lediglich den allein genutzten Rechner des Beschuldigten betrifft, kann es unverhältnismäßig sein, den gesamten gespeicherten Datenbestand zu sichern,

der in aller Regel zu großen Teilen für das Strafverfahren ohne Belang sein wird. Soweit möglich müssen daher bereits im Rahmen der Durchsuchung verfahrenserhebliche und -unerhebliche Daten getrennt und Sicherungsmaßnahmen auf den relevanten Teil beschränkt werden. Zudem muss so weitgehend wie möglich gewährleistet werden, dass keine Daten gesichert werden, die dem unantastbaren Kernbereich der persönlichen Lebensgestaltung zuzuordnen sind.

Die zuvor dargestellten Sicherungsmethoden, mit denen der Umfang der Sicherung bereits vor Ort begrenzt werden kann, können daher mildere Mittel im Sinne des Verhältnismäßigkeitsgrundsatzes darstellen, so dass es rechtlich geboten sein kann, sie einzusetzen. Die Schwierigkeit besteht jedoch praktisch immer darin, eine inhaltlich unbekannte Menge an Daten auf einen für das Ermittlungsverfahren relevanten Datenbestand zu reduzieren. Dies ist nur möglich, indem Informationen über die Datenmenge und ihren Inhalt gewonnen werden. Die Frage ist, ob es zum Zeitpunkt der Sicherung möglich ist, eine ausreichende Informationslage herzustellen, auf Basis derer begründete und nachvollziehbare Entscheidungen zur Reduktion der zu sichernden Daten getroffen werden können. Am Ende kommt es auf die Umstände jedes einzelnen Verfahrens an: Was wird gesucht? Wie schwer wiegt der Tatvorwurf? Welche Ressourcen stehen den Ermittlern zur Verfügung? Diese Fragen müssen im Kontext beantwortet werden und bestimmen letztendlich die Sicherungsstrategie.

8.3 Organisation und Analyse großer Datenmengen

Die sichergestellten Datenmengen steigen seit Jahren und stellen die Ermittler in der Praxis vor große Herausforderungen. Insbesondere die Aufbereitung und Analyse großer Datenmengen stellt die Ermittler vor weitere Probleme: Wie, durch wen, in welchen Räumlichkeiten und in welchem Zeitraum soll die Sichtung erfolgen? Diese Fragen können nur im konkreten Kontext beantwortet und sollen in diesem Abschnitt2 aus Sicht der aktuellen Praxis diskutiert werden. Hierbei gibt es starke Bezüge zu den Vorgehensmodellen aus Kapitel 5.

Der Bereich der Datenaufbereitung und Datenanalyse wird in der Forschung häufig mit ***„computational forensics“*** und ***„e-discovery“*** bezeichnet und berührt andere Bereiche der Informatik wie Datenbanken und künstliche Intelligenz, die nicht primär eine forensische Perspektive einnehmen.

8.3.1 Priorisierung der Daten

Zunächst müssen die Daten für den gesamten nachfolgenden Prozess der Aufbereitung und Auswertung priorisiert werden. Die Priorisierung geschieht in der Regel zunächst nach Personen. Beispielsweise erhält der Rechner des Hauptbeschuldigten eine höhere Priorität als der Rechner eines nur am Rande involvierten Zeugen. Diese Prioritäten ergeben sich somit aus dem Lastenheft der Sachermittler. Weiterhin können mehrere Rechner oder Datenträger einer Person nach ihrer Wichtigkeit priorisiert werden. Beispielsweise können Vermutungen darüber angestellt werden, welcher Rechner am häufigsten genutzt wurde. Ferner werden Daten häufig nach der Schwierigkeit ihrer Erhebung geordnet. Eine einfache intakte Festplatte wird beispielsweise meist vor einem defekten RAID-Verbund analysiert werden.

Ein wichtiger Aspekt bei der Priorisierung von Daten ist, dass es nicht aufgrund der Priorisierung dazu kommen darf, dass ein niedrig priorisiertes Beweismittel nicht untersucht wird, weil zuvor bereits hinreichend belastendes Material gefunden wurde. In der Praxis hingegen gibt es durchaus ein Ermittlungsziel. Nach Bestätigung dieses Ermittlungsziels kann mit dem Verteidiger des Beschuldigten darüber verhandelt werden, ob er auf der vollständigen Auswertung der bislang noch nicht gesichteten Daten besteht, da er davon ausgeht, dass noch entlastende Informationen zu finden sind oder ob die für die vollständige Analyse anfallenden Kosten (dem Beschuldigten) erspart werden können. Meist kann sich in einem solchen Fall sogar direkt auf ein geeignetes Strafmaß geeinigt werden. Zur Bezifferung der für die vollständige Analyse der Daten anfallenden Kosten kann der Aufwand, der von einem externen Dienstleister in Rechnung gestellt würde, als Ausgangspunkt dienen. Eine solche Einigung muss in den meisten Fällen jedoch auch mit dem Einbehalt der sichergestellten Geräte einhergehen, so dass diese gegebenenfalls auch später für eine weitere Analyse zur Verfügung stehen.

Sollten sich auf den Geräten mutmaßlich illegale Daten, wie beispielsweise kinderpornographisches Material, befinden, dürfen diese auf keinen Fall ohne eine vollständige Analyse und Säuberung zurückgegeben werden, da die ermittelnden Behörden sich andernfalls selbst der Verbreitung dieses Materials schuldig machen würden.

8.3.2 Aufbereitung von Massendaten

Die Aufbereitung von in Hinsicht auf ihre Quantität und Heterogenität komplexen Datenmengen stellt eine der häufigsten Aufgaben im Bereich der

digitalen Forensik im weiteren Sinne dar. Die nun beschriebene Vorgehensweise steht im Zusammenhang mit allgemeinen Vorgehensmodellen, wie sie auch in Kapitel 5 beschrieben werden. Sie lässt sich in die folgenden sechs Schritte unterteilen:

1. Erhebung
2. Expansion
3. Aggregation
4. Reduktion
5. Strukturierung
6. Visualisierung

Wir betrachten nun diese in Abbildung 8.2 auf der nächsten Seite schematisch dargestellten Schritte im Einzelnen.

Erhebung

In einem ersten Schritt werden zunächst alle Daten gesammelt und inhaltlich unbekannte Daten strukturiert. Beispielsweise muss festgestellt werden, welche Daten sich überhaupt auf einem Datenträger befinden, also welche Partitionen existieren und mit welchen Dateisystemen diese formatiert wurden. Auf höheren Abstraktionsebenen muss geklärt werden, um welche Arten von Daten es sich konkret handelt, also ob beispielsweise komplexe Datenbanken, virtuelle Maschinen oder bestimmte Dateitypen vorliegen. Selbstverständlich wird im Rahmen der Datenerhebung auch versucht, gelöschte und versteckte Daten wiederherzustellen. Schwierig wird es in der Praxis, wenn wichtige RAID-Systeme lediglich mit Hilfe der Informationen auf den Festplatten wiederhergestellt werden müssen (man denke auch an Software-RAIDs), da hier die Daten über mehrere physische Festplatten verteilt gespeichert sind. Ebenfalls als schwierig erweisen sich häufig Zugriffe auf proprietäre Datenbanken, von denen möglicherweise kein Schema vorhanden oder dieses zumindest nicht bekannt ist.

Expansion

Die im ersten Schritt gesammelten Daten müssen unter Umständen noch expandiert werden. Beispielsweise können Archive entpackt werden oder einzelne E-Mails aus Postfächern und E-Mail-Anhänge aus den E-Mails extrahiert werden. Wo dies möglich ist, kann auch versucht werden, verschlüsselte Dokumente zu entschlüsseln. Einfach ist dies meist bei nur schwach geschützten Dokumenten oder Archiven. Bei einem verschlüsselten Daten-Container, der mit aktueller Software mit einem hohen Sicherheitsstandard erstellt wurde, gestaltet sich dies jedoch schwierig.

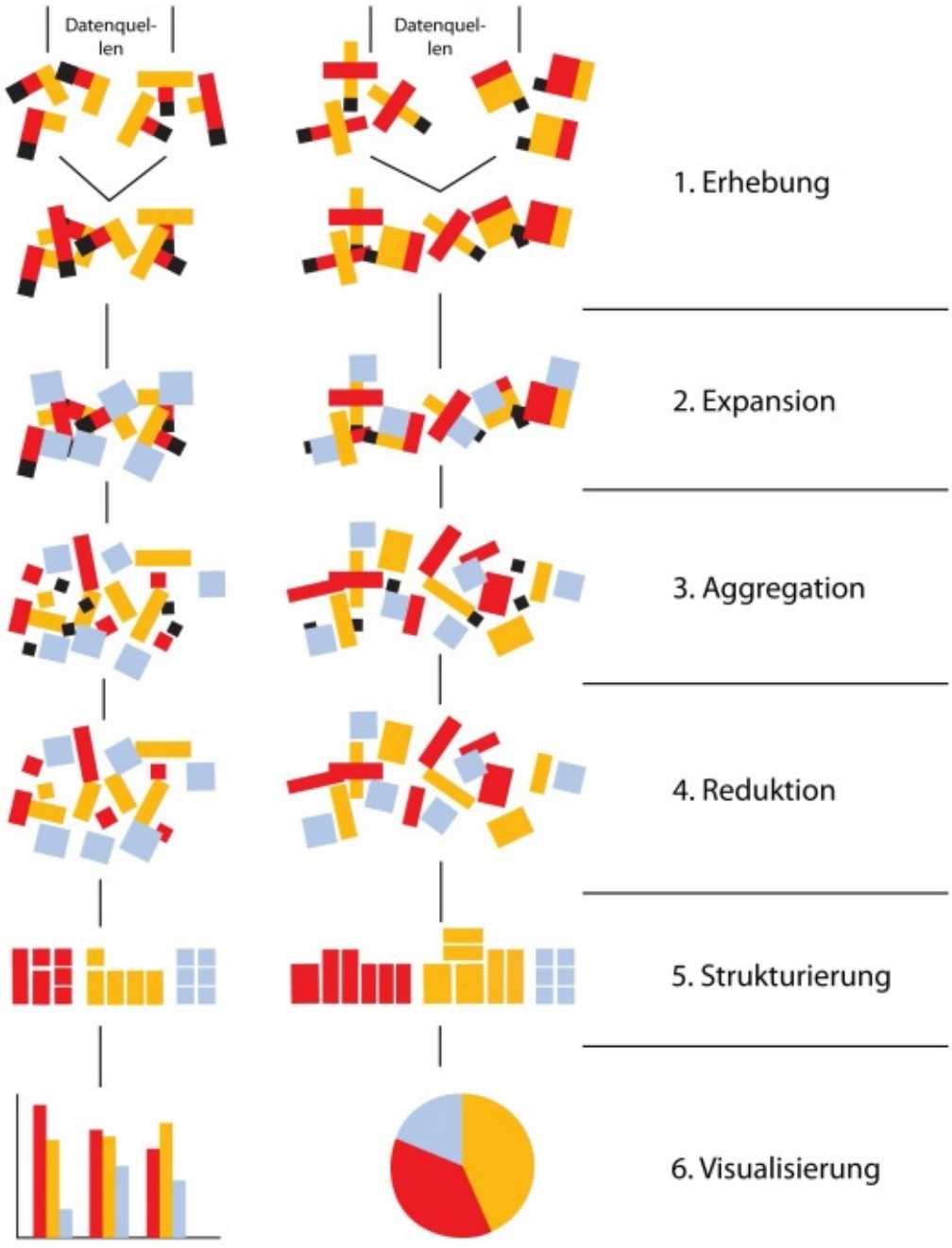


Abbildung 8.2: Schematische Darstellung der sechs Schritte zur Aufbereitung von Daten.

Aggregation

Im nächsten Schritt werden die unterschiedlichen Inhalte zusammengeführt. Hierbei spielen neben technischen Herausforderungen (wie unterschiedlichen Kodierungsverfahren) die verschiedenen Arten von Daten eine Rolle. Beispielsweise unterscheiden sich Dokumente, E-Mails und Chat-Logs sehr stark hinsichtlich ihrer Strukturen und Metadaten. Außerdem liegen meist gleichartige Daten in unterschiedlichen Formaten vor, die in der Regel von der verwendeten Software abhängen. Im Optimalfall können all diese Daten mit ihren Metadaten in eine gemeinsame Datenbank eingepflegt werden, um die im Folgenden beschriebenen Arbeitsschritte zu erleichtern. Bei dieser Homogenisierung der Daten müssen häufig kleine Skripte und Parser eingesetzt werden, um Daten von einem Format in ein anderes zu konvertieren, oder die Ausgabe eines Programmes weiterzuverarbeiten.

Reduktion

Aufgabe der Reduktion ist es, auf Basis der im vorgenannten Schritt erstellten Datenbank redundante Daten aus dem Datenbestand herauszufiltern. Dieser Schritt wird vor allem relevant, wenn aus mehreren der zusammengeführten Datenquellen die gleichen Daten vorliegen, wie es beispielsweise bei Vorliegen mehrerer Sicherungen (Backups) der Fall ist. Durch die große Überlappung der Daten in einem solchen Fall kann die zu sichtende Datenmenge und damit der Arbeitsaufwand signifikant reduziert werden. Dieser häufig auch als De-Duplizierung bezeichnete Schritt erfordert die Definition geeigneter Einzigartigkeits- bzw. Gleichheitskriterien in Bezug auf die vorliegenden Daten. Häufig wird als Übereinstimmungskriterium die Hashwert-Gleichheit verwendet. In einigen Fällen ist es jedoch sinnvoll, dieses exakte Kriterium durch eigene, weichere aber semantisch sinnvolle Kriterien zu ersetzen.

Strukturierung

Die vorliegenden Inhalte müssen nun strukturiert und durchsuchbar gemacht werden. Hierzu bieten sich gängige Indizierungsmechanismen sowie

entsprechende Suchalgorithmen an. Zu beachten ist, welche Art der Benutzerschnittstelle im konkreten Fall zur Verfügung gestellt werden soll. Während in einigen Fällen lediglich automatisierte Anfragen über Kommandozeilenparameter an die Suchmaschine gestellt werden sollen, müssen in anderen Fällen eventuell technisch weniger versierte Ermittler die Schnittstelle nutzen, um nach konkreten Beweismitteln zu fahnden. Weiterhin kann die Suche lokal auf Arbeitsrechnern der Ermittler oder auf einem zentralen Server durchgeführt werden. Auch Features wie die Möglichkeit, Kommentare zu gesichteten Dateien für andere Ermittler zu den Daten hinzuzufügen, können hier die Arbeit erleichtern. Insbesondere in zivilrechtlichen Verfahren dürfen aufgrund datenschutzrechtlicher Beschränkungen jedoch nicht alle Daten gesichtet werden. In diesem Fall müssen Filter eingesetzt werden, die lediglich das Sichten für das Verfahren relevanter Daten erlauben.

Visualisierung

Zuletzt müssen die Daten visualisiert werden. Zur inhaltlichen Auswertung können auch Virtualisierungslösungen zum Einsatz kommen. Dies ermöglicht das Sichten der eventuell durch eine Suchanfrage gefundenen Daten in ihrem Originalzustand mit Hilfe der jeweils auf dem Ursprungssystem eingesetzten Client-Anwendung. Diese Anwendungen bieten häufig auch interne Such- und Filtermöglichkeiten, so dass in kleinen Fällen der Schritt der Strukturierung unter Umständen entfallen kann. Zur Analyse komplexerer Datenformate wie Logdateien oder Datenbanken kommt in der Regel spezielle Software zum Einsatz. Abgesehen vom Anzeigen der gefundenen Daten kann es durchaus sinnvoll sein, allein bestimmte Metadaten zu visualisieren. Beispiele hierfür können die Anzahl von Dateien eines bestimmten Formates oder die statistische Auswertung der Verteilung von Rechnungsbeträgen sein, um Unregelmäßigkeiten festzustellen.

Diese Aufgabenliste ist nicht vollständig. Sie lässt sich durchaus weiter ergänzen und birgt eine Menge Detailfragen. Auch sind, wie zum Teil bereits angedeutet, nicht immer alle Schritte erforderlich bzw. sinnvoll. Sie geben aber einen Rahmen für eine Ermittlung mit großen Datenmengen vor. Es bleibt noch einmal hervorzuheben, dass für all die zuvor genannten Aufgaben zur Automatisierung Skripte und Parser eingesetzt werden sollten, um die Arbeitsbelastung der Ermittler zu senken. Teilweise gibt es hierfür auch

Standard-Tools: Insbesondere unter dem Stichwort ***e-discovery*** finden sich eine Menge entsprechender Software-Lösungen.

8.4 Zusammenfassung

In diesem Kapitel wurden einige praktische Aspekte digitaler Ermittlungen behandelt. Die Darstellung basiert auf den Erfahrungen von Praktikern, die im Rahmen von strukturierten Interviews befragt wurden. Außerdem wurde auf den praktisch relevanten Bereich des Umgangs mit Massendaten eingegangen und es wurden zwei komplementäre Lösungsstrategien besprochen: Einerseits die selektive Sicherung, andererseits die strukturierte Expansion und De-Duplizierung.

Literaturverzeichnis

[American National Standards Institute 1994] AMERICAN NATIONAL STANDARDS INSTITUTE: ***American National Standard for Information Systems: AT Attachement Interface for Disk Drivers; ANSI X3.221-1994***. ANSI, 1994 (American National Standard) Zitiert auf Seite 190.

[American National Standards Institute 1998] AMERICAN NATIONAL STANDARDS INSTITUTE: ***American National Standard for Information Systems: AT Attachmentwith Packet Interface Extension (ATA/ATAPI-4); ANSI NCITS 317-1998***. ANSI, 1998 (American National Standard) Zitiert auf Seite 185.

[American National Standards Institute 2000] AMERICAN NATIONAL STANDARDS INSTITUTE: ***American National Standard for Information Systems: AT Attachmentwith Packet Interface Extension (ATA/ATAPI-5); ANSI NCITS 340-2000***. ANSI, 2000 (American National Standard) Zitiert auf Seite 191.

[American National Standards Institute 2002] AMERICAN NATIONAL STANDARDS INSTITUTE: ***American National Standard for Information Systems: AT Attachmentwith Packet Interface Extension (ATA/ATAPI-6); NCITS 361-2002***. ANSI, 2002 (American National Standard) Zitiert auf den Seiten 184, 186 und 190.

[Amerini u. a. 2011] AMERINI, Irene ; BALLAN, Lamberto ; CALDELLI, Roberto ; BIMBO, Alberto D. ; SERRA, Giuseppe: A SIFT-based Forensic Method for Copy-Move Attack Detection and Transformation Recovery. In: ***IEEE Transactions on Information Forensics and Security*** 6 (2011), September, Nr. 3, S. 1099–1110 Zitiert auf Seite 143.

[Apache Software Foundation 2011] APACHE SOFTWARE FOUNDATION: ***Apache Subversion***. http://subversion.apache.org/, Juli 2011 Zitiert auf Seite 223.

[Bäcker u. a. 2010] BÄCKER, Matthias ; FREILING, Felix C. ; SCHMITT, Sven: Selektion vor der Sicherung - Methoden zur effizienten forensischen Sicherung von digitalen Speichermedien. In: ***Datenschutz und Datensicherheit - DuD*** 34 (2010), Nr. 2, S. 80–85. – ISSN 1614–0702. – 10.1007/s11623-010-0040-4 Zitiert auf den Seiten 55 und 248.

[Barroso 2015] BARROSO, Ariadne: ***SATA cable***. http://www.publicdomainpictures.net/view-image.php?image=64093. Version: 2015 Zitiert auf Seite 185.

[Baryamureeba u. Tushabe 2004] BARYAMUREEBA, V. ; TUSHABE, F.: The Enhanced Digital Investigation Process Model. In: ***DFRWS***, 2004 Zitiert auf Seite 178.

[Beebe u. Clark 2005] BEEBE, Nicole ; CLARK, Jan G.: A hierarchical, objectives-based framework for the digital investigations process. In: ***Digital Investigation*** 2 (2005), Nr. 2, 147–167. http://dx.doi.org/10.1016/j.diin.2005.04.002 Zitiert auf Seite 178.

[Beligum 2009] BELIGUM: ***ScreenKast***. http://sourceforge.net/projects/screenkast/, Juli 2009. – Version 0.1.4 Zitiert auf Seite 219.

[Benecke 2001] BENECKE, Mark: ***Genetischer Fingerabdruck***. ecomed Verlagsgesellschaft, 2001. – Enzyklopädie Naturwissenschaft und Technik, 2. Auflage, 6. Ergänzungslieferung Zitiert auf Seite 26.

[Billard u. Hauri 2010] BILLARD, David ; HAURI, Rolf: Making sense of unstructured flash-memory dumps. In: ***Proceedings of the 2010 ACM Symposium on Applied Computing***. New York, NY, USA : ACM, 2010 (SAC ’10). – ISBN 978–1–60558–639–7, 1579–1583 Zitiert auf Seite 183.

[Böhme u. a. 2009] BÖHME, Rainer ; GLOE, Thomas ; FREILING, Felix C. ; KIRCHNER, Matthias: Multimedia forensics is not computer forensics. In: ***Proceedings of the 3rd International Workshop on Computational Forensics***. Den Haag, 2009, S. 90–103 Zitiert auf Seite 24.

[Brodowski u. a. 2014] BRODOWSKI, Dominik ; DEWALD, Andreas ; FREILING, Felix C. ; KOVÁCS, Steve ; RIEGER, Martin: Drei Jahre Master Online Digitale Forensik: Ergebnisse und Erfahrungen. In: ***SICHERHEIT — Schutz und Zuverlässigkeit. Tagung des Fachbereichs Sicherheit der Gesellschaft für Informatik***. Wien, 2014 Zitiert auf Seite 4.

[Brodowski u. Freiling 2011] BRODOWSKI, Dominik ; FREILING, Felix C.: ***Cyberkriminalität, Computerstrafrecht und die digitale Schattenwirtschaft***. Berlin : Forschungsforum Öffentliche Sicherheit, 2011 (Schriftenreihe Sicherheit 4) Zitiert auf Seite 1.

[Böhme u. a. 2009] BÖHME, Rainer ; FREILING, Felix C. ; GLOE, Thomas ; KIRCHNER, Matthias: Multimedia Forensics Is Not Computer Forensics. In: ***Proceedings of the 3rd International Workshop on Computational Forensics***. Berlin, Heidelberg : Springer-Verlag, 2009 (IWCF '09). – ISBN 978–3–642–03520–3, S. 90–103 Zitiert auf Seite 56.

[Carrier 2005] CARRIER, Brian: ***File System Forensic Analysis***. Addison-Wesley, 2005 Zitiert auf den Seiten 3, 37, 38, 40, 42, 55, 72, 172, 182, 184, 188, 191, 195, 196, 197, 198, 201, 204, 207, 210 und 215.

[Carrier u. Spafford 2003] CARRIER, Brian D. ; SPAFFORD, Eugene H.: Getting Physical with the Digital Investigation Process. In: ***Int. Journal of Digital Evidence*** 2 (2003), Nr. 2. http://www.utica.edu/academic/institutes/ecii/publications/articles/A0AC5AFB6C-325D-BF515A44FDEE7459.pdf Zitiert auf den Seiten 59, 177 und 178.

[Carrier u. Spafford 2004] CARRIER, Brian D. ; SPAFFORD, Eugene H.: Defining event reconstruction of digital crime scenes. In: ***Journal of Forensic Science*** 49 (2004), Nr. 6 Zitiert auf den Seiten 56 und 60.

[Carvey u. Altheide 2005] CARVEY, Harlan ; ALTHEIDE, Cory: Tracking USB storage: Analysis of windows artifacts generated by USB storage devices. In: ***Digital Investigation*** 2 (2005), Nr. 2, S. 94–100 Zitiert auf den Seiten 47 und 48.

[Casey 2004] CASEY, Eoghan: ***Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet***. Academic Press, 2004. – 2. Auflage Zitiert auf den Seiten 1, 2, 29, 151, 153, 157, 158 und 165.

[Casey 2011] CASEY, Eoghan: ***Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet***. Academic Press, 2011. – 3. Auflage Zitiert auf den Seiten ix, 35, 52, 53, 54, 56, 60 und 224.

[Chandy u. Misra 1988] CHANDY, K. Mani ; MISRA, J.: ***Parallel program design***. Boston, MA, USA : Addison-Wesley Pub. Co. Inc., 1988. – ISBN 0–201–05866–9 Zitiert auf Seite 66.

[Cheddad u. a. 2010] CHEDDAD, Abbas ; CONDELL, Joan ; CURRAN, Kevin ; KEVITT, Pauf M.: Digital Image Steganography: Survey and Analysis of Current Methods. In: ***Signal Processing*** 90 (2010), März, Nr. 3, S. 727–752 Zitiert auf den Seiten 118, 120 und 122.

[Chen u. a. 2008] CHEN, Mo ; FRIDRICH, Jessica J. ; GOLJAN, Miroslav ; LUKÁS, Jan: Determining Image Origin and Integrity Using Sensor Noise. In: ***IEEE Transactions on Information Forensics and Security*** 3 (2008), Nr. 1, S. 74–90 Zitiert auf den Seiten 46 und 47.

[Cohen 2010] COHEN, Fred: Toward a Science of Digital Forensic Evidence Examination. In: ***IFIP Int. Conf. Digital Forensics***, 2010, S. 17–35 Zitiert auf Seite 60.

[Cohen 2011] COHEN, Fred: ***Digital Forensic Evidence Examination***. ASP Press, 2011 Zitiert auf Seite 60.

[Cohen u. a. 2011] COHEN, Fred ; LOWRIE, Julie ; PRESTON, Charles: The State of the Science of Digital Evidence Examination. In: ***IFIP Int. Conf. Digital Forensics***, 2011, S. 3–21 Zitiert auf Seite 60.

[Compaq Computer Corporation 1996] COMPAQ COMPUTER CORPORATION, Intel Corporation Phoenix Technologies Ltd.: ***BIOS Boot Specification:Version 1.01***, January 1996 Zitiert auf Seite 187.

[Cox u. a. 1997] Cox, Ingemar J. ; KILIAN, Joe ; LEIGHTON, F. T. ; SHAMOON, Talal: Secure Spread Spectrum Watermarking for Multimedia. In: ***IEEE***

Transactions on Image Processing 6 (1997), Dezember, Nr. 12, S. 1673–1687 Zitiert auf den Seiten 125, 126 und 127.

[Dardick u.a. 2014] DARDICK, Glenn S. ; ENDICOTT-POPOVSKY, Barbara ; GLADYSHEV, Pavel ; KEMMERICH, Thomas ; RUDOLPH, Carsten: Digital Evidence and Forensic Readiness (Dagstuhl Seminar 14092). In: ***Dagstuhl Reports*** 4 (2014), Nr. 2, 150–190. http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0030-drops-45490. – URN urn:nbn:de:0030–drops–45490. – ISSN 2192–5283 Zitiert auf Seite 33.

[Department of Justice 2002] DEPARTMENT OF JUSTICE: ***Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations***. Internet: http://www.cybercrime.gov/s&smanual2002.pdf und http://www.cybercrime.gov/s&smanual2002.htm, 2002 Zitiert auf Seite 159.

[Department of Justice u. National Institute of Justice 2001] DEPARTMENT OF JUSTICE ; NATIONAL INSTITUTE OF JUSTICE: ***Electronic Crime Scene Investigation: A Guide to First Responders***. Internet: http://www.ncjrs.gov/pdffiles1/nij/187736.pdf, 2001 Zitiert auf den Seiten 159, 160, 161, 162, 163 und 164.

[Dewald 2012] DEWALD, Andreas: ***Formalisierung digitaler Spuren und ihre Einbettung in die Forensische Informatik***, Friedrich-Alexander-Universität Erlangen- Nürnberg, Diss., 2012. – http://opus4.kobv.de/opus4-fau/files/2741/AndreasDewaldDissertation.pdf Zitiert auf Seite 104.

[Dewald u. Freiling 2012] DEWALD, Andreas ; FREILING, Felix: Is Computer Forensics a Forensic Science? In: ***Proceedings of Current Issues in IT Security 2012*** Max-Planck-Institut für ausländisches und internationales Strafrecht, Universität Freiburg, 2012 Zitiert auf Seite 60.

[Dewald u. Freiling 2014] DEWALD, Andreas ; FREILING, Felix:: From Computer Forensics to Forensic Computing: Investigators Investigate, Scientists Associate / Friedrich-Alexander-University Erlangen-Nuremberg (FAU). 2014 (CS-2014-04). –Forschungsbericht. – Online: https://opus4.kobv.de/opus4-

fau/files/4750/computer_forensics_is_not_forensic_science.pdf Zitiert auf Seite 60.

[Dewald u. a. 2010] DEWALD, Andreas ; FREILING, Felix C. ; WEBER, Tim: Design and Implementation of a Documentation Tool for Interactive Command Line Sessions / Uni Mannheim. 2010 (TR-2010-005). – Forschungsbericht. – Quelltext verfügbar unter https://github.com/scy/forscript Zitiert auf Seite 219.

[Dijkstra 1975] DIJKSTRA, Edsger W.: Guarded commands, nondeterminacy and formal derivation of programs. In: ***Commun. ACM*** 18 (1975), August, S. 453–457. – ISSN 0001–0782 Zitiert auf Seite 66.

[Dornseif 2004] DORNSEIF, Maximillian: ***Vorlesung Computerforensik***. 2004 Zitiert auf den Seiten xi, 157 und 159.

[Duranti 1998] DURANTI, Luciana: ***Diplomatics: New Uses for an Old Science***. Scarecrow Press, 1998 Zitiert auf Seite 33.

[Epiphan Systems Inc. 2011] EPIPHAN SYSTEMS INC.: ***Recording Products***. http://www.epiphan.com/products/recording, September 2011 Zitiert auf Seite 219.

[Farid 2011] FARID, Hany: ***Digital Image Forensics***. http://www.cs.dartmouth.edu/farid/downloads/tutorials/digitalimageforensic Juni 2011 Zitiert auf den Seiten 137, 138 und 148.

[Farmer u. Venema 2005] FARMER, Dan ; VENEMA, Wietse: ***Forensic Discovery***. Addison-Wesley, 2005 Zitiert auf den Seiten 2, 60 und 61.

[Freiling u. Gruhn 2015] FREILING, Felix C. ; GRUHN, Michael: What is Essential Data in Digital Forensic Analysis? In: ***Ninth International Conference on IT Security Incident Management & IT Forensics, IMF 2015, Magdeburg, Germany, May 18-20, 2015***, 2015, 40–48 Zitiert auf Seite 38.

[Freiling u. Schwittay 2007] FREILING, Felix C. ; SCHWITTAY, Bastian: A Common Process Model for Incident Response and Computer Forensics. In: FRINGS, Sandra (Hrsg.) ; GÖBEL, Oliver (Hrsg.) ; GÜNTHER, Detlef

(Hrsg.) ; HASE, Hardo (Hrsg.) ; NEDON, Jens (Hrsg.) ; SCHADT, Dirk (Hrsg.) ; BRÖMME, Arslan (Hrsg.): ***IT-Incidents Management & IT-Forensics - IMF 2007, Conference Proceedings, September 11-13, 2007, Stuttgart, Germany*** Bd. 114, GI, 2007 (LNI). – ISBN 978–3–88579–208–6, S. 19–40 Zitiert auf den Seiten 151 und 166.

[Fridrich 2000] FRIDRICH, Jiri: Steganalysis of LSB Encoding in Color Images. In: ***IEEE International Conference on Multimedia and Expo*** Bd. 3, 2000, S. 1279–1282 Zitiert auf Seite 121.

[Gallagher u. Chen 2008] GALLAGHER, A. ; CHEN, T.: Image Authentication by Detecting Traces of Demosaicing. In: ***Computer Vision and Pattern Recognition Workshops***, 2008, S. 1–8 Zitiert auf Seite 144.

[Garfinkel 2006] GARFINKEL, Simson L.: AFF: a new format for storing hard drive images. In: ***Commun. ACM*** 49 (2006), Nr. 2, 85–87. http://doi.acm.org/10.1145/1113034.1113076 Zitiert auf Seite 212.

[Garfinkel 2010] GARFINKEL, Simson L.: Digital Forensics Research: The Next 10 Years. In: ***Proceedings of the Digital Forensics Research Conferencs (DFRWS***), 2010 Zitiert auf Seite 56.

[Garfinkel u. a. 2006] GARFINKEL, Simson L. ; MALAN, David J. ; DUBEC, Karl-Alexander ; STEVENS, Christopher C. ; PHAM, Cecile: AFF: An Open Extensible Format for Disk Imaging. In: OLIVIER, Martin S. (Hrsg.) ; SHENOI, Sujeet (Hrsg.): ***Advances in Digital Forensics II - IFIP International Conference on Digital Forensics, National Centre for Forensic Science, Orlando, Florida, USA, January 29 - February 1, 2006*** Bd. 222, Springer, 2006 (IFIP Advances in Information and Communication Technology). – ISBN 978–0–387–36890–0, 13–27 Zitiert auf Seite 212.

[Geschonneck 2006] GESCHONNECK, Alexander: ***Computer Forensik***. dpunkt Verlag, 2006. – 2. Auflage Zitiert auf den Seiten 1, 2, 55, 59 und 61.

[Giloi u. Lauber 1963] GILOI, W. ; LAUBER, R.: ***Analogrechnen – Programmierung, Arbeitsweise und Anwendung des elektronischen Analogrechners***. Springer-Verlag, 1963 Zitiert auf Seite 29.

[Gimp 2012] GIMP, GNU: ***GNU Image Manipulation Program***. http://www.gimp.org, April 2012 Zitiert auf Seite 120.

[Git Development Team 2011] GIT DEVELOPMENT TEAM: ***Git – Fast Version Control System***. http://git-scm.com/, Juli 2011 Zitiert auf Seite 223.

[Gloe u. a. 2010] GLOE, T. ; BOROWKA, K. ; WINKLER, A.: Efficient Estimation and Large-scale Evaluation of Lateral Chromatic Aberration for Digital Image Forensics. In: ***SPIE Media Forensics and Security*** Bd. 2, 2010, S. 7541—7547 Zitiert auf Seite 134.

[Gollmann 2011] GOLLMANN, Dieter: ***Computer Security***. Third. Wiley, 2011. – I–XIX, 1–436 S. http://www.wiley-vch.de/publish/dt/books/ISBN978-0-470-74115-3/. – ISBN 978–0–470–74115–3 Zitiert auf den Seiten 31 und 32.

[Gong u. a. 2005] GONG, Ruibin ; CHAN, Tony Kai Y. ; GAERTNER, Mathias: Case-Relevance Information Investigation: Binding Computer Intelligence to the Current Computer Forensic Framework. In: ***Int. Journal of Digital Evidence*** 4 (2005), Nr. 1. http://www.utica.edu/academic/institutes/ecii/publications/articles/B4A6A1(A93D-85B1-96C575D5E35F3764.pdf Zitiert auf Seite 178.

[Grance u. a. 2004] GRANCE, Tim ; KENT, Karen ; KIM, Brian: Computer Security Incident Handling Guide / National Institute of Standards and Technology (NIST). Version: Januar 2004. http://csrc.nist.gov/publications/nistpubs/800-66/sp800-61.pdf. 2004 (SP 800-61). – Special Publication Zitiert auf Seite 154.

[Groß u. Geerds 1977] GROSS, Hans ; GEERDS, Friedrich: ***Handbuch der Kriminalistik***. Bd. 1. Verlagsgesellschaft Manfred Pawlak, 1977 Zitiert auf den Seiten 8, 13 und 16.

[Gärtner u. Völzer 2000] GÄRTNER, Felix C. ; VÖLZER, Hagen: Redundancy in space in fauft-tolerant systems / Technische Universität Darmstadt. 2000 (TUD-BS-2000-006). – Forschungsbericht Zitiert auf Seite 69.

[Hayati u.a. 2007] HAYATI, Pedram ; POTDAR, Vidyasagar ; CHANG, Elizabeth: A Survey of Steganographic and Steganalytic Tools for the Digital Forensic Investigator. In: ***Workshop on Information Hiding and Digital Watermarking***, 2007 Zitiert auf Seite 148.

[Hopcroft u. a. 2002] HOPCROFT, John ; MOTWANI, Rajeev ; ULLMAN, Jeffrey: ***Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie***. Addison-Wesley Longman, 2002 Zitiert auf Seite 64.

[Huffman u. Pless 2003] HUFFMAN, Cary W. ; PLESS, Vera: ***Fundamentals of Error-Correcting Codes***. Cambridge University Press, 2003. – ISBN 0521782805 Zitiert auf Seite 69.

[Inman u. Rudin 2000] INMAN, Keith ; RUDIN, Norah: ***Principles and Practice of Criminalistics: The Profession of Forensic Science***. CRC Press, 2000 Zitiert auf den Seiten 11, 12, 18, 19, 20, 21, 24, 26, 45, 56, 60 und 63.

[James u. Nordby 2009] JAMES, Stuart H. (Hrsg.) ; NORDBY, Jon J. (Hrsg.): ***Forensic Science: An Introduction to Scientific and Investigative Techniques***. Third. CRC Press, 2009 Zitiert auf Seite 59.

[Johnson u. Farid 2006] JOHNSON, M. ; FARID, H.: Exposing Digital Forgeries through Chromatic Aberration. In: ***ACM Workshop on Multimedia and Security***, 2006, S. 48–55 Zitiert auf Seite 132.

[Johnson u. Farid 2007] JOHNSON, M. ; FARID, H.: Exposing Digital Forgeries in Complex Lighting Environments. In: ***IEEE Transactions on Information Forensics and Security*** 2 (2007), September, Nr. 3, S. 450–461 Zitiert auf Seite 146.

[Kent u. a. 2006] KENT, K. ; CHEVALIER, S. ; GRANCE, T. ; DANG, H.: Guide to Integrating Forensics into Incident Response / Computer Security Division Information Technology Laboratory, National Institute of Standards and Technology. 2006 (Special Publication 800-86). – Forschungsbericht. – http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf Zitiert auf Seite 179.

[Kharrazi u. a. 2004] KHARRAZI, Mehdi ; SENCAR, Husrev T. ; MEMON, Nasir: Blind Source Camera Identification. In: ***IEEE International Conference on Image Processing***, 2004, S. 709–712 Zitiert auf Seite 134.

[Kiltz u. a. 2009] KILTZ, Stefan ; HOPPE, Tobias ; DITTMANN, Jana ; VIELHAUER, Claus: Video surveillance: A new forensic model for the forensically sound retrival of picture content off a memory dump. In: FISCHER, Stefan (Hrsg.) ; MAEHLE, Erik (Hrsg.) ; REISCHUK, Rüdiger (Hrsg.): ***Informatik 2009: Im Focus das Leben, Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 28.9.-2.10.2009, Lübeck, Proceedings*** Bd. 154, GI, 2009 (LNI). – ISBN 978–3–88579–248–2, S. 1619–1633 Zitiert auf den Seiten 179 und 180.

[Kirchner 2010] KIRCHNER, Matthias: Linear Row and Column Predictors for the Analysis of Resized Images. In: ***ACM SIGMM Multimedia & Security Workshop***, 2010, S. 13–18 Zitiert auf Seite 144.

[Kirk 1974] KIRK, Paul L.: ***Crime Investigation***. John Wiley & Sons, 1974. – Herausgegeben von John I. Thornton, 2. Auflage Zitiert auf den Seiten 13, 16, 17, 20, 21, 22, 23 und 24.

[Kornblum 2006] KORNBLUM, Jesse: Identifying Almost Identical Files Using Context Triggered Piecewise Hashing. In: ***Digit. Investig***. 3 (2006), September, 91–97. http://dx.doi.org/10.1016/j.diin.2006.06.015. – DOI 10.1016/j.diin.2006.06.015. – ISSN 1742–2876 Zitiert auf Seite 214.

[Kruse II u. Heiser 2001] KRUSE II, Warren G. ; HEISER, Jay G.: ***Computer Forensics: Incident Response Essentials***. Boston, MA, USA : Addison-Wesley Pub. Co. Inc., 2001. – ISBN 978–0201707199 Zitiert auf Seite 60.

[Lee u. Harris 2000] LEE, Henry C. ; HARRIS, Howard A.: ***Physical Evidence in Forensic Science***. Lawyers and Judges Publishing, 2000 Zitiert auf Seite 16.

[Li u. a. 2011] Li, Bin ; HE, Junhui ; HUANG, Jiwu ; SHI, Yun Q.: A Survey on Image Steganography and Steganalysis. In: ***Journal of Information Hiding and Multimedia Signal Processing*** 2 (2011), April, Nr. 2, S. 142–172 Zitiert auf den Seiten 119, 121 und 122.

[Locard 1920] LOCARD, Edmund: ***L'enquete criminelle et les methodes scientifique***. Ernest Flammarion, Paris, 1920 Zitiert auf Seite 18.

[Lowe 2004] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: ***International Journal of Computer Vision*** 60 (2004), November, Nr. 2, S. 91–110 Zitiert auf Seite 143.

[Lukáš u. Fridrich 2003] LUKÁŠ, J. ; FRIDRICH, J.: Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In: ***Digital Forensics Research Workshop***, 2003 Zitiert auf Seite 140.

[Lukáš u. a. 2006] LUKÁŠ, Jan ; FRIDRICH, Jessica ; GOLJAN, Miroslav: Digital Camera Identification From Sensor Pattern Noise. In: ***IEEE Transactions on Information Forensics and Security*** 1 (2006), Juni, Nr. 2, S. 205–214 Zitiert auf den Seiten 135 und 137.

[Lukás u. a. 2005] LUKÁŠ, Jan ; FRIDRICH, Jessica J. ; GOLJAN, Miroslav: Digital "bullet scratches" for images. In: ***ICIP (3***), 2005, S. 65–68 Zitiert auf Seite 47.

[Lynch 2000] LYNCH, Clifford A.: Authenticity and Integrity in the Digital Environment: An Exploratory Analysis of the Central Role of Trust. In: COUNCIL ON LIBRARY AND INFORMATION RESOURCES (Hrsg.): ***Authenticity in a Digital Environment***. Washington, D.C., 2000, S. 32–50. – Online: http://www.clir.org/pubs/reports/reports/pub92/pub92.pdf Zitiert auf den Seiten 32 und 33.

[Mairgiotis u. Galatsanos 2010] MAIRGIOTIS, Antonis ; GALATSANOS, Nikolaos: Bayesian Watermark Detection and New Perceptual Mask Based on a Spatially Weighted Total Variation Image Prior. In: ***IEEE Workshop on Information Forensics and Security***, 2010 Zitiert auf Seite 125.

[Mandia u. a. 2003] MANDIA, Kevin ; PROSISE, Chris ; PEPE, Matt: ***Incident Response & Computer Forensics***. McGraw-Hill, 2003. – 2. Auflage Zitiert auf den Seiten 55, 56, 151, 154, 155 und 156.

[Margot 2011] MARGOT, Pierre: Forensic science on trial — What is the law of the land? In: ***Australian Journal of Forensic Sciences*** 43 (2011), Nr. 2–

3, 89–103. http://dx.doi.org/10.1080/00450618.2011.555418 Zitiert auf Seite 10.

[Mathworks-Deutschland 2012] MATHWORKS-DEUTSCHLAND: ***MATLAB***. http://www.mathworks.de/products/matlab, April 2012 Zitiert auf Seite 148.

[Menezes u. a. 1997] MENEZES, Alfred J. ; OORSCHOT, Paul C. V. ; VANSTONE, Scott A.: ***Handbook of Applied Cryptography***. CRC Press, 1997 Zitiert auf den Seiten 31, 32 und 41.

[Nikkel 2009] NIKKEL, Bruce J.: Forensic analysis of GPT disks and GUID partition tables. In: ***Digital Investigation*** 6 (2009), Nr. 1-2, 39–47. http://dx.doi.org/10.1016/j.diin.2009.07.001 Zitiert auf Seite 201.

[NIST 2011] NIST: ***NIST National Software Reference Library***. Internet: http://www.nsrl.nist.gov, September 2011 Zitiert auf Seite 165.

[Nölle 2009] NÖLLE, Rolf: ***Herodot Historien***. B o D — Books on Demand, 2009 Zitiert auf Seite 118.

[Octave 2012] OCTAVE, GNU: ***Octave***. http://www.gnu.org/software/octave, April 2012 Zitiert auf Seite 148.

[OpenCV 2012] OPENCV: ***Open Computer Vision Library***. http://opencv.willowgarage.com, April 2012 Zitiert auf Seite 149.

[Overill u. Silomon 2012] OVERILL, Richard E. ; SILOMON, Jantje A. M.: Uncertainty Bounds for Digital Forensic Evidence and Hypotheses. In: ***Seventh International Conference on Availability, Reliability and Security, Prague, ARES 2012, Czech Republic, August 20-24, 2012***, 2012, 590–595 Zitiert auf Seite 54.

[Overill u. a. 2010] OVERILL, Richard E. ; SILOMON, Jantje A. M. ; CHOW, Kam-Pui: A Complexity Based Model for Quantifying Forensic Evidential Probabilities. In: ***ARES 2010, Fifth International Conference on Availability, Reliability and Security, 15-18 February 2010, Krakow, Poland***, 2010, 671–676 Zitiert auf Seite 54.

[Overill u. a. 2013] OVERILL, Richard E. ; SILOMON, Jantje A. M. ; CHOW, Kam-Pui ; TSE, Hayson: Quantification of digital forensic hypotheses using probability theory. In: ***Eighth International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2013, Hong Kong, China, November 21-22, 2013***, 2013, 1–5 Zitiert auf Seite 54.

[Pan u. Lyu 2010] PAN, Xunyu ; LYU, Siwei: Region Duplication Detection Using Image Feature Matching. In: ***IEEE Transactions on Information Forensics and Security*** 5 (2010), Dezember, Nr. 4, S. 857–867 Zitiert auf Seite 143.

[Papula 2006] PAPULA, Lothar: ***Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler***. 6. Vieweg+Teubner Verlag, 2006 Zitiert auf Seite 122.

[Paul 2009] PAUL, George L.: ***Foundations of Digital Evidence***. Chicago, USA : American Bar Association, 2009. – ISBN 978–1604421040 Zitiert auf den Seiten 60 und 61.

[Pérez-González u. Hernández 1999] PÉREZ-GONZÁLEZ, Fernando ; HERNÁNDEZ, Juan R.: A Tutorial on Digital Watermarking. In: ***IEEE International Carnahan Conference on Security Technology***, 1999, S. 286–292 Zitiert auf den Seiten 123 und 124.

[Pollitt 2007] POLLITT, Mark: An Ad Hoc Review of Digital Forensic Models. In: HUANG, Ming-Yuh (Hrsg.) ; FRINCKE, Deborah A. (Hrsg.): ***Second International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2007, Seattle, Washington, USA, April 10-12, 2007***, IEEE Computer Society, 2007. – ISBN 0–7695–2808–2, 43–54 Zitiert auf Seite 175.

[Pollitt 2008] POLLITT, Mark: Applying traditional forensic taxonomy to digital forensics. In: RAY, Indrajit (Hrsg.) ; SHENOI, Sujeet (Hrsg.): ***Advances in Digital Forensics IV***. Springer Verlag, 2008, Kapitel 2, S. 17–27 Zitiert auf Seite 60.

[Pollitt 1995] POLLITT, Mark M.: Computer Forensics: An Approach to Evidence in Cyberspace. In: ***Proc. 18th NIST-NCSC National Information***

Systems Security Conference, 1995, S. 487–491 Zitiert auf Seite 175.

[Popescu u. Farid 2005a] POPESCU, A. ; FARID, H.: Exposing Digital Forgeries by Detecting Traces of Resampling. In: ***Signal Processing*** 53 (2005), Februar, Nr. 2, S. 758–767 Zitiert auf den Seiten 144 und 145.

[Popescu u. Farid 2005b] POPESCU, A. ; FARID, H.: Statistical Tools for Digital Forensics. In: ***Information Hiding Conference***, 2005, S. 395–407 Zitiert auf Seite 140.

[Popper 1962] POPPER, Karl: ***Conjectures and Refutations: The Growth of Scientific Knowledge***. Routledge, 1962. – Deutsch: Vermutungen und Widerlegungen Zitiert auf den Seiten 6 und 61.

[Regan 2009] REGAN, James E.: ***The Forensic Potential of Flash Memory***. Monterey, CA, Naval Postgraduate School, Diplomarbeit, 2009. http://handle.dtic.mil/100.2/ADA509258. – 86 S. Zitiert auf Seite 183.

[Rogers 2013] ROGERS, Corinne: Digital Records Forensics: Integrating Archival Science into a General Model of the Digital Forensics Process. In: ***Proc. of the Second International Workshop on Cyberpatterns: Unifying Design Patterns with Security, Attack and Forensic Patterns*** Oxford Brookes University, 2013, S. 4–21. – http://tech.brookes.ac.uk/CyberPatterns2013/ Zitiert auf Seite 33.

[Rogers u. Seigfried 2003] ROGERS, Marcus K. ; SEIGFRIED, Kate: The future of computer forensics: a needs analysis survey / Center for Education and Research in Information Assurance and Security (CERIAS). Purdue University, 2003 (2003-30). – Forschungsbericht Zitiert auf Seite 1.

[Ryu u. a. 2010] RYU, S. ; LEE, M. ; LEE, H.: Detection of Copy-Rotate-Move Forgery using Zernike Moments. In: ***Information Hiding Conference***, 2010, S. 51–65 Zitiert auf Seite 142.

[Sachnev u. a. 2009] SACHNEV, Vasiliy ; KIM, Hyoung J. ; ZHANG, Rongyue: Less Detectable JPEG Steganography Method Based on Heuristic Optimization and BCH Syndrome Coding. In: ***Proceedings of the 11th***

ACM Workshop on Multimedia and Security, 2009, S. 131–140 Zitiert auf Seite 120.

[Saferstein 2010] SAFERSTEIN, Richard: ***Criminalistics: An Introduction to Forensic Science***. 10th. Pearson, 2010 Zitiert auf Seite 59.

[Schüle 2008] SCHÜLE, Christian: Die Unsichtbare. In: ***Die Zeit*** (2008), April, Nr. 18 Zitiert auf Seite 54.

[Schurich 1974] SCHURICH, F.-R.: Zur Definition des Begriffes "Spur". In: ***Kriminalistik und forensische Wissenschaften*** (1974), Nr. 14, S. 5–27 Zitiert auf Seite 9.

[Schurich 1982] SCHURICH, F.-R.: Kriminalistische Identifizierung und angrenzende Begriffe. In: ***Kriminalistik und forensische Wissenschaften*** (1982), Nr. 46, S. 7–22 Zitiert auf Seite 9.

[Schurich u. Scharf 1979] SCHURICH, F.-R. ; SCHARF, H.: Zum Begriff der Widerspiegelung in der sozialistischen Kriminalistik. In: ***Kriminalistik und forensische Wissenschaften*** (1979), Nr. 36, S. 5–21 Zitiert auf Seite 9.

[Spiegel 2010] ***Der Spiegel***. Juni 2010. – No. 23 Zitiert auf den Seiten 142 und 143.

[S.Tanenbaum 1987] S.TANENBAUM, Andrew: ***Operating Systems: Design and Implementation***. Prentice-Hall, 1987. – ISBN 0–13–637331–3 Zitiert auf den Seiten 187 und 198.

[Stüttgen u. a. 2013] STÜTTGEN, Johannes ; DEWALD, Andreas ; FREILING, Felix: Selective Imaging Revisited. In: SIDAR, GI (Hrsg.): ***Proceedings of the 7th International Conference on IT Security Incident Management & IT Forensics***, 2013, S. 0–0 Zitiert auf Seite 211.

[Tardos 2003] TARDOS, Gábor: Optimal Probabilistic Fingerprint Codes. In: ***ACM symposium on Theory of Computing***, 2003, S. 116–125 Zitiert auf Seite 128.

[TechSmith Corporation 2011] TECHSMITH CORPORATION: ***Camtasia Studio***. http://www.techsmith.de/, September 2011 Zitiert auf Seite 219.

[Unified EFI, Inc. 2013] UNIFIED EFI, INC.: ***Unified Extensible Firmware Interface Specification***. 2.4, June 2013. http://www.uefi.org/specs/download Zitiert auf den Seiten 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208 und 209.

[Van u. a. 2007] VAN, Lanh T. ; EMMANUEL, Sabu ; KANKANHALLI, Mohan S.: Identifying Source Cell Phone using Chromatic Aberration. In: ***IEEE International Conference on Multimedia and Expo***, 2007, S. 883–886 Zitiert auf Seite 134.

[Varouhakis 2008] VAROUHAKIS, John: ***recordMyDesktop***. http://recordmydesktop.sourceforge.net/, Dezember 2008. – Version 0.3.8.1 Zitiert auf Seite 219.

[Wei u. a. 2011] WEI, Michael ; GRUPP, Laura M. ; SPADA, Frederick E. ; SWANSON, Steven: Reliably Erasing Data from Flash-based Solid State Drives. In: ***Proceedings of the 9th USENIX Conference on File and Stroage Technologies***. Berkeley, CA, USA : USENIX Association, 2011 (FAST'11). – ISBN 978–1–931971–82–9, 8–8 Zitiert auf Seite 183.

[Weihmann 2007] WEIHMANN, Robert: ***Kriminaltechnik I***. VDP, 2007. – Band 2 der Lehr- und Studienbriefe Kriminalistik/Kriminologie, dritte Auflage. Zitiert auf den Seiten 9 und 13.

[Weihmann 2008] WEIHMANN, Robert: Versionsbildung als Unterdrückungsinstrument der DDR-Diktatur. In: ***Kriminalistik*** (2008), Nr. 1, S. 28–31 Zitiert auf Seite 9.

[Wikipedia 2008] WIKIPEDIA: ***Chs.svg (svg image)***. https://commons.wikimedia.org/wiki/File:Chs.svg. Version: 2008 Zitiert auf Seite 199.

[Wikipedia 2010] WIKIPEDIA: ***PATA-cable.jpg (JPG image)***. https://en.wikipedia.org/wiki/File:PATA-cable.jpg. Version: 2010 Zitiert auf Seite 184.

[Wikipedia 2015] WIKIPEDIA: ***Cylinder Head Sector (svg image)***. https://en.wikipedia.org/wiki/File:Cylinder_Head_Sector.svg. Version:

2015 Zitiert auf Seite 190.

[Zimmermann u. a. 2012] ZIMMERMANN, Christian ; SPREITZENBARTH, Michael ; SCHMITT, Sven ; FREILING, Felix: Forensic Analysis of YAFFS2. In: GI (Hrsg.): ***Proceedings of GI Sicherheit 2012***, 2012 Zitiert auf Seite 183.

Index

1:1-Kopie, →

A Study in Scarlet, →
Action, →
almost certain (Stufe C5), →
Annahme, →
Anthropometrie, →
Anzahl der Partitionstabelleneinträge, →
Asservatenbeutel, →
Asservatenkammer, →
Assoziation, →, →, →
assumption, →
asymetrisches Verschlüsselungsverfahren, →
Auswerteantrag, →
Authentizität, →, →, →

Behauptung, →–→, →
Benutzer-Sekundärdaten, →
Bertillon, Alphonse, →
Beweismittel, →, →
big data, →
Block, →
Blutgruppen, →
boot code, →
Boot-Indikator, →
Boot-Signatur, →
Breite (einer Untersuchung), →

certain (Stufe C6), →
chain loading bootloader, →
chain of custody, →, →, →, →, →, →

changed variables, →
Characteristic Counter Evidence, →
Characteristic Evidence, →
CHS End-Adresse, →
CHS Start-Adresse, →
CHS-Adressierung, →
chunks, →
claim, →
classification, →
clone copy, →
closed under subsets, →
Covering Set, →
Command, →
Common Characteristic Counter Evidence, →
Common Characteristic Evidence, →
common model, →
computational forensics, →
Computer Emergency Response Teams, →
computer security incident, →
computer security incident response teams, →
Conan Doyle, Sir Arthur, →
Coordinated Universal Time, →
Counter Evidence, →
Cover Channel, →
Cover-Verteilung, →
CRC32-Prüfsumme der Partitionstabelle, →
criminalistics, →
cylinder, →

Daktyloskopie, →
data analysis, →
data collection, →
data mining, →
Dateiebene, →
Dateisysteme, →
Dateisystemebene, →
Datenschutztechniken, →

datenzentrisches Vorgehensmodell, →
dd, →
DDR, →
De-Duplizierung, →
dead analysis, →
device instance identifier, →
digest, →
digital crime scene, →, →
digital evidence, →, →
digitale Signatur, →
digitale Spuren, →, →
digitalen Tatort, →
diplomatics, →
disk image, →
diskrete Repräsentation digitaler Spuren, →
diskrete Zustände, →
divisibility, →
DNA-Fingerabdruck, →
Domain, →
downtime, →
Dupin, Auguste, →

e-discovery, →, →
EFI-Signatur, →
eindeutig unterscheidbare Zustände, →
embedded image, →
EnCase, →
End-LBA der Partition, →
Equivalent Set, →
Ereignis, →, →, →
Erfahrungsfalle, →
Erlaubter Eingriff, →
Ermittler, →
erroneous (Stufe C0), →
erweiterte Partitionen, →
essential data, →
essentielle Daten, →

event, →
events, →
evidence, →
evidence bag, →
Evidence Set, →
exclusively used variables, →
explizite Konfigurationsdaten, →

fehlerhaft, →
Festplattenabbild, →
Festplattengeometrie, →
Final State, →
Fingerabdruck, →
finite, →
flüchtige Spuren, →
Flat Fielding, →
forensic science, →
Forensik, →
forensisch, →
forensische Informatik, →
Forensische Nutzung, →
forum, →
Fragile Wasserzeichen, →
fruit of the poisonous tree, →
FTK Imager, →

Galton, Francis, →
General Reconstruction Problem, →
Geräteinstanzkennung, →
Geräteidentifikation, →
geschlossenes System, →, →
Gröse einer Partition, →
Gröse eines Partitionstabelleneintrages, →
Grade von Wahrscheinlichkeit, →
Groß, Hans, →, →
Guard, →

Handbuch der Kriminalistik, →

Hardware Write Blocker, →
hardware write blocker, →, →
hash, →
Hashfunktion, →, →
Hashwert, →
Hauptspeicheranalyse, →
Herkunft, →
highly uncertain (Stufe C1), →
hinterlassenes Zeichen, →
Holmes, Sherlock, →, →
Hypothese vs. Version, →
Hypothesen, →

identification, →
Identifizierung, →, →
imaging, →, →
implizite Konfigurationsdaten, →
incident response, →
Incident-Response-Modell, →
incorrect (Stufe C0), →
individualisierende Merkmale, →
Individualisierung, →, →, →
individualization, →
inferior, →
Information, →
information, →
Inhaltsauthentifizierung, →
Initial State, →
Initial Value, →
inkorrekt, →
inkorrekt (Stufe C0), →
Integrität, →, →, →
Interference, →
Interfering Actions, →
investigative Prozess, →
IT-Sicherheitsvorfall, →

Jeffreys, Sir Alex, →

Kapazität, →
Kapazität von LSB, →
Klassifizierung, →, →
Konfigurationsdaten, →
Kontakt, →
kontrollierter Herstellungsprozess, →, →
Kriminalistik, →
Kriminalistik, sozialistische, →
Kriminaltaktik, →
Kriminaltechnik, →
kriminaltechnische Untersuchung, →, →
Kriminologie, →
kryptographische Hashfunktionen, →, →

Landsteiner, Karl, →
Laufwerk, →
LBA, →
LBA des ersten Blocks der Partition Area, →
LBA Start Adresse, →
levels of certainty, →
Live-Analyse, →, →
Locard's exchange principle, →
Locard, Edmund, →
Locards Austauschprinzip, →, →
Logdaten, →
logical block address, →
logical partition volume address, →
logical volume address, →
logische Adressierung, →
logische Blockadresse, →
logische Laufwerksadresse, →
logische Partitionen, →
logische Partitionsadresse, →
LSB-Einbettung, →

möglich, →

möglich (Stufe C3), →
marques multiples, →
Massendatenanalyse, →
Maske, →
Merged Evidence, →
mikroskopische Spuren, →
minderwertig, →
Minimal Covering Set, →
Minimal Equivalent Set, →
Modus Operandi, →
multi stage bootloader, →
Multimediaforensik, →

Nachprüfbarkeit, →
Nachvollziehbarkeit, →
Netzwerk-Fingerprinting, →
Netzwerkforensik, →
Non-Interference, →

Objektivität, →
Order Of Volatility, OOV, →
Organisation der Verbrechensbekämpfung, →

partially essential data, →
Partiell essentielle Daten, →
Partition, →
partition, →
Partitionen, →
Partitions-GUID, →
Partitionsebene, →, →
Partitionsnamen, →
Partitionssystem, →
Partitionstabelle, →
Partitionstyp, →
Partitionstyp-GUID, →
perfekte Kopien, →
perfekte Sicherheit, →
perfektes Verbrechen, →

persistente Spuren, →
physical crime scene, →
physical evidence, →
physical transfer, →
physische Adressierung, →
physische Ebene, →
physische Spuren, →
Pixelnachbarschaft, →
Platten, →
platters, →
Poe, Edgar Allen, →
possible (Stufe C3), →
Post-Mortem-Analyse, →
Primärdaten, →
primäre Dateisystempartition, →
primäre erweiterte Partition, →
primary extended partition, →
primary file system partition, →
private key, →
PRNU-Rauschen, →
probable (Stufe C4), →
Program, →
Programmcode, →
Programmdaten, →
provenance, →
public key, →

raw image, →
read-only mounted device, →
Rechenaufwand, →
Rechenschieber, →
Registry, →
Rekonstruktion, →
response posture, →
Reverse Engineering, →
Robuste Wasserzeichen, →
Robustheit, →

Robustheit LSB, →

Sachbeweise, →
secondary extended partition, →
secondary file system partition, →
secret key, →
sectors, →
sehr unwahrscheinlich, →
sehr unwahrscheinlich (Stufe C1), →
sehr wahrscheinlich, →
sehr wahrscheinlich (Stufe C5), →
Seitenkanalanalyse, →
Sektoren, →
Sekundärdaten, →
sekundäre Dateisystempartition, →
sekundäre erweiterte Partition, →
selective ***imaging***, →
selektives Sichern, →
semi-persistente Spuren, →
sicher, →
sicher (Stufe C6), →
Sicherheit LSB, →
sniffing, →
Software Write Blocker, →
solid state drives, →
somewhat uncertain (Stufe C2), →
Specific Group Reconstruction Problem, →
Specific Reconstruction Problem, →
Spur, →, →
Spuren, →
Spurenanalyse, →
Spurenherkuft, →
Spureninformation, →, →
Spurensicherung, →
Spurenträger, →, →, →
Stand der Wissenschaft, →
Stand des Berichts, →

Start-LBA der Partition, →
State, →
Stego-Verteilung, →
strictly essential data, →
Strikt essentielle Daten, →
Strong Interference, →
support, →
supreme court, →
System-Sekundärdaten, →

Tableau Imager, →
technisch unvermeidbare Spuren, →
technisch vermeidbare Spuren, →
The Red-Headed League, →
The Tragedy of Pudd'nhead Wilson, →
Tiefe (einer Untersuchung), →
Tot-Analyse, →
Total Interference, →
Träger, →
Trace, →
tracks, →
Transfer, →, →
transfer of matter, →
transfer of traits, →, →, →
transiente Spuren, →
Triage, →
Twain, Mark, →

Übertragung von Materie, →
Übertragung von Mustern, →, →
Uhlenbuth, Pauf, →
Unentdeckbarkeit, →
Unerlaubter Eingriff, →
unkontrollierter Herstellungsprozess, →
unwahrscheinlich, →
unwahrscheinlich (Stufe C2), →
Urkundenwesen, →

UTC, →

Variables, →
Verbrechenstechnik, →
Verschwörer, →
Version, →
Versionsbildung, →
Versionshistorie, →
Verwahrungskette, →, →
Verwertbarkeitsverbot, →
volume, →
Vučetić, Ivan, →

wahrscheinlich, →
wahrscheinlich (Stufe C4), →
Weak Interference, →
wear leveling, →

X-Ways, →

Zeit des Untersuchungsobjekts, →
Zeitstempel, →
zero evidence, →
Zerteilbarkeit, →
Ziele der Bildforensik, →
Zone Bit Recording-Technik, →
zufälliger Herstellungsprozess, →, →
Zylinder, →

2. Auflage, Oktober 2015

Bibliographische Information der Deutschen Bibliothek:
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie;
Detaillierte bibliographische Daten sind
im Internet über http://dnb.ddb.de abrufbar.

Gesetzt durch die Autoren
Herstellung: Books on Demand GmbH, Norderstedt

ISBN 9783732212088