Vorlesung Datensicherheit

WS 2003/2004

Dr. Frank Bourseau

Dr. Jens Fricke

Personalia

- Dr. Frank Bourseau
- Studium der Mathematik und Physik
- Diplom und Promotion in Mathematik an der Universität Bielefeld
- Danach Berater im Datenbankbereich
- Seit 4 1/2 Jahren im IT-Sicherheitsbereich
- Seit 1. Jan. 2002 bei der dvg Hannover, jetzige FinanzIT

Personalia

- Dr. Jens Fricke
- Studium der Physik und Mathematik
- Diplom und Promotion in Physik an der Universität Göttingen
- Seit 1998 Systementwickler und IT-Sicherheitsberater
- Seit 1. Feb. 2002 bei der dvg Hannover, jetzige FinanzIT

Inhalt der Vorlesung (1)

- Kapitel 1: Sicherheitsmanagement
  - Bedrohungen und Risiken
  - Standards und Vorgehensweisen
- Kapitel 2: Einige Bedrohungen
  - Viren, Würmer, Trojaner
  - Buffer Overflow
  - Angriffe auf Web-Anwendungen

Inhalt der Vorlesung (2)

- Kapitel 3: Kryptographie
  - Symmetrische Kryptographie
  - Hash Algorithmen, Message Digests
  - Public Key Kryptographie
  - Schlüssel-Management
- Kapitel 4: Netzwerksicherheit
  - Überblick über TCP/IP
  - Schwachstellen und Bedrohungen IP-basierter Protokolle und Dienste
  - Sichere elektronische Kommunikation
  - Firewalls

Inhalt der Vorlesung (3)

- Kapitel 5: Authentifizierung
  - Passwörter
  - Token
  - Biometrische Verfahren
  - Kerberos
- Kapitel 6: Drahtlose Kommunikation
  - Wireless LAN
  - Bluetooth

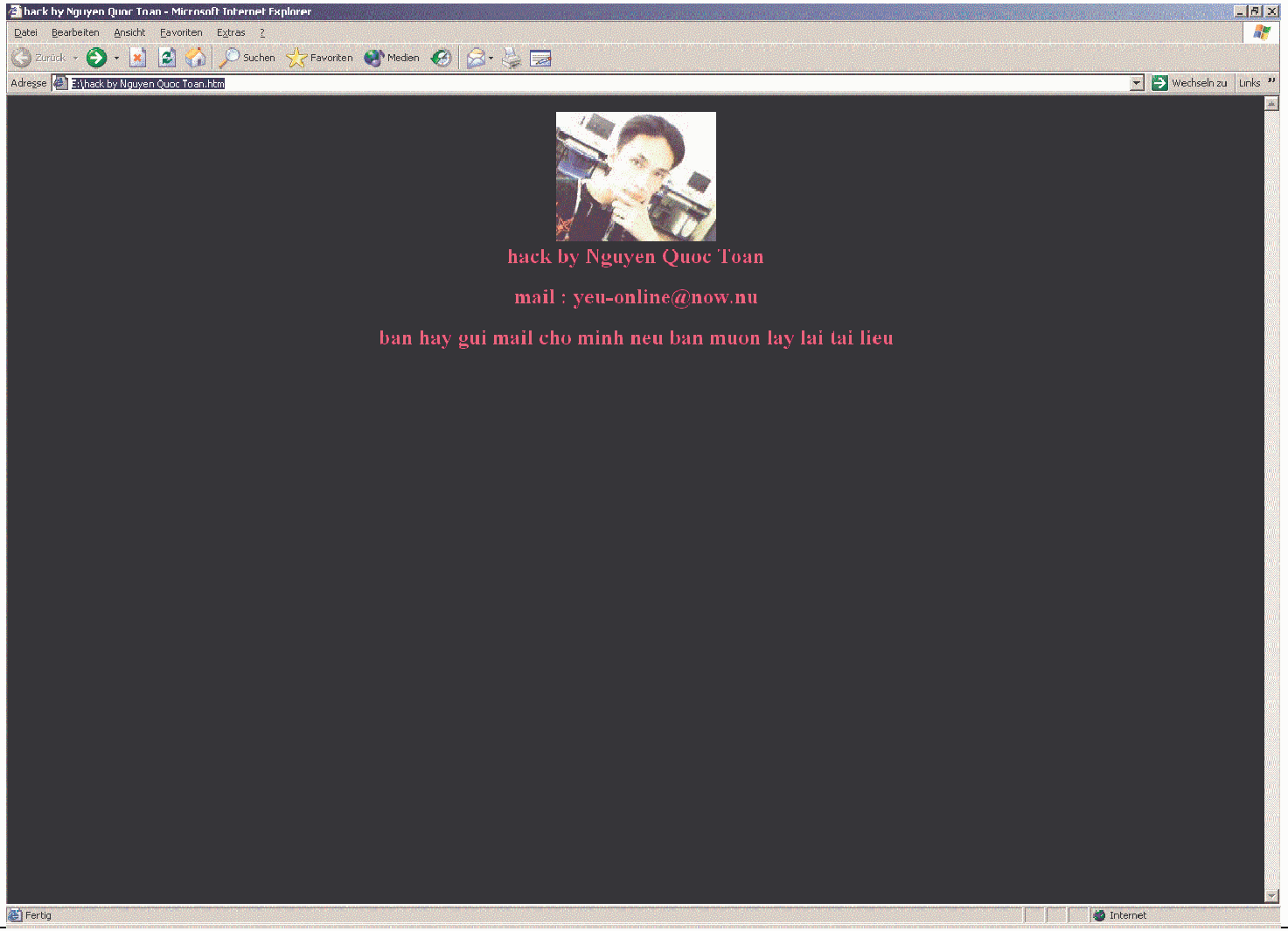
Sicherheitsmanagement

- Kapitel 1: Sicherheitsmanagement
  - Bedrohungen und Risiken
  - Standards und Vorgehensweisen

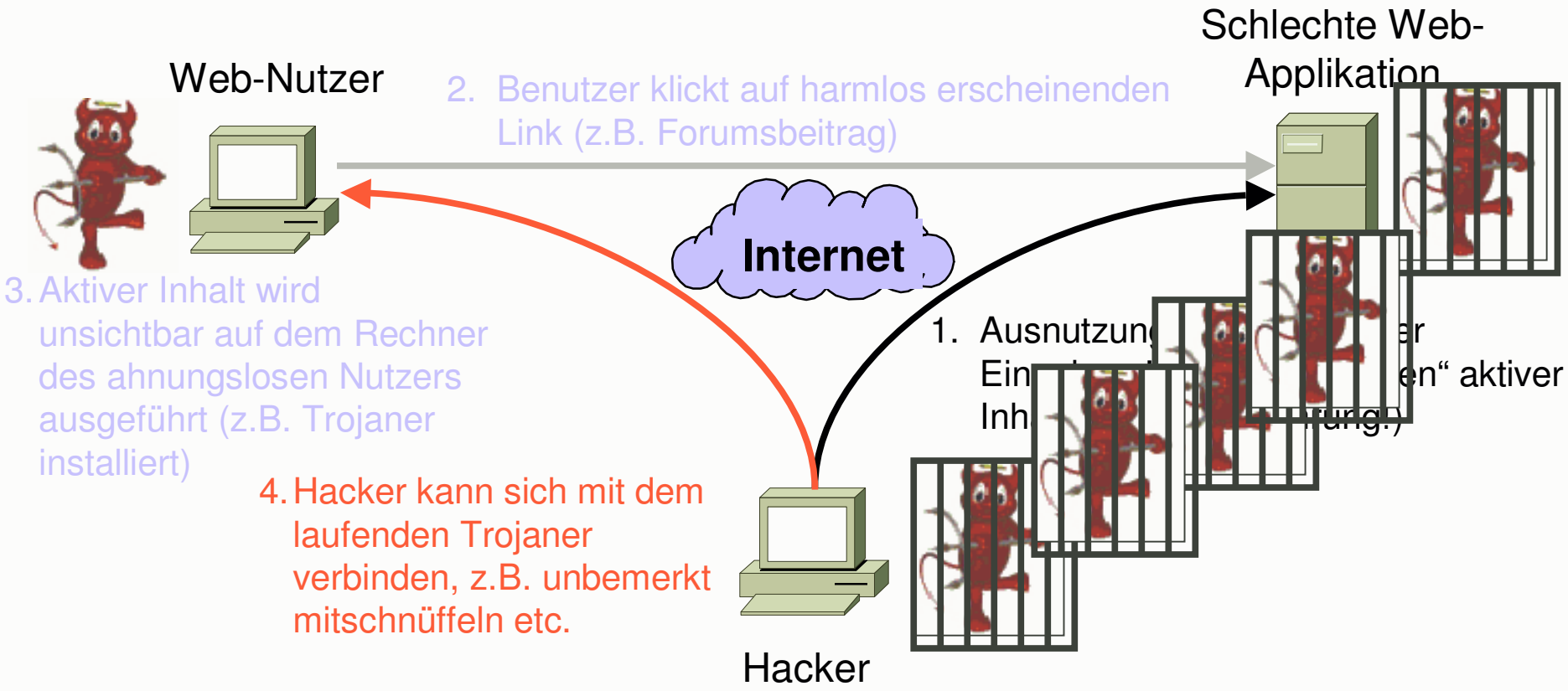
Heise News Ticker 23.10.02

Heise News Ticker 23.10.02

www.klickbilderbox.de am 19.10.03



Bsp.: Cross-Site-Scripting



Begriffe (1)

- Informationssicherheit (Information Security)
  - Allgemein: Schutz vor der (bewussten oder unbewussten) Verletzung der Integrität, Authentizität, Vertraulichkeit oder Verfügbarkeit von Informationen
- IT-Sicherheit (IT-Security):
  - Informationssicherheit bei elektronischer Datenverarbeitung bzw. –speicherung
- Datensicherung:
  - Schutz vor Informationsverlust

Begriffe (2)

- Notfallplanung (Recovery):
  - Planung von Reaktions- und Wiederanlaufszenarien bei Ausfall wichtiger Geschäftsbestandteile, wie Technik, Informationen, Gebäude, Mitarbeiter
- Datenschutz (Privacy):
  - Schutz vor unzulässiger Verarbeitung und Nutzung personenbezogener Informationen
- Betriebssicherheit (Safety)
  - Schutz vor Schäden an Leib und Leben

- Datenschutz bedient sich IT-Sicherheitsmaßnahmen!
- IT-Sicherheitsmaßnahmen können zu Datenschutzrechten im Widerspruch stehen.

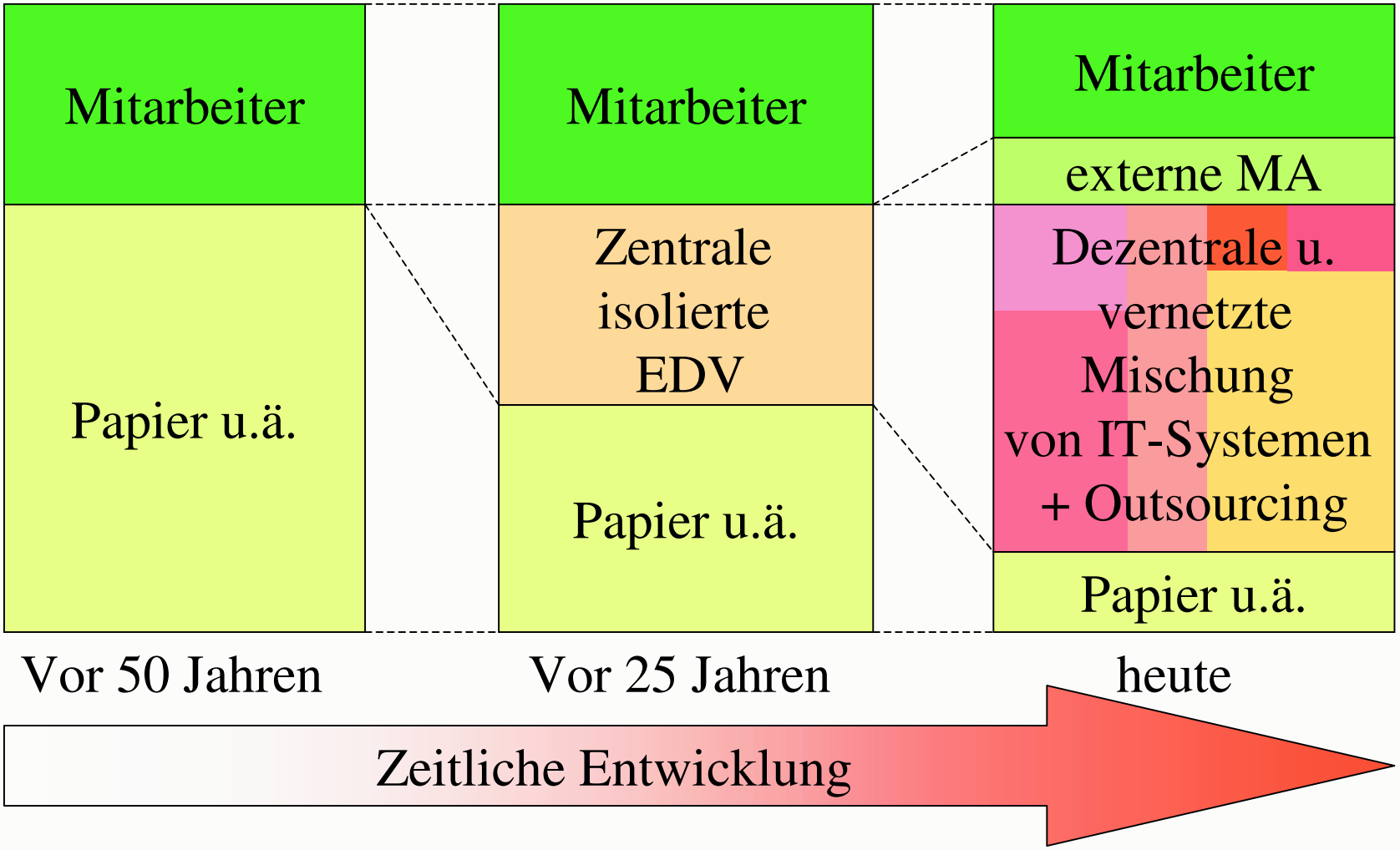
Was ist Sicherheit? (1)

- Ist es sicher zu fliegen?
- Jedes komplexe System hat Schwachstellen (Cockpittüren offen).
- Bedrohung: Flugzeugentführung
- Risiko = Bedrohung x Wahrscheinlichkeit für Eintritt

Was ist Sicherheit? (2)

- Subjektive Größe
- Vom Betrachter abhängig
- Nicht direkt sicht- oder messbar
- Idealerweise bleibt sie unbemerkt.
- Schwer zu „verkaufen“
- 100% Sicherheit nicht erreichbar

Informationsverteilung



Die Herausforderungen

- Sicherheitsadministration
- Zunahme von technischen Sicherheitslösungen
- Rechte- und Rollenverwaltung
- Verschiedenste Plattformen
- Zunahme von Schnittstellen
- Externe und interne Prüfungen
- Gesetzliche Anforderungen
- Finanzielle Bewertung von Sicherheitsmaßnahmen, Kostenfaktor?
- Risikoabschätzungen
- ...

Thesen zur IT-Unsicherheit

- IT-Systeme und Kommunikationsstrukturen werden ständig komplexer.
- Komplexe Systeme sind für Menschen kaum beherrschbar.
- Resultierende Schwachstellen werden bewusst oder unbewusst ausgenutzt oder führen systemimmanent zu Sicherheitsverletzungen.
- Die Abhängigkeit von funktionierender IT nimmt ständig zu.
- Sicherheit in einem IT-System ist bis heute selten ein Auswahlkriterium.
- Mehr Sicherheitstechnik lindert die Symptome, erhöht aber auch die Komplexität.
- „Technische Mindest-Sicherheitsstandards“ veralten so schnell wie die dazugehörigen Infrastrukturen.
- **Ausweg:**
- Etablierung eines Informationssicherheitsmanagementsystems (ISMS).
- Lernen, die Komplexität besser zu beherrschen und, wo möglich, abzubauen.

IT-Sicherheit

- Sicherheit der Informationstechnik?
- Was ist das schützenswerte Gut, die Informationstechnik?
- Nein: Zu schützen sind Informationen und zwar vor Diebstahl, Veränderung, Abstreitbarkeit, etc.
- Da viele Informationen innerhalb von Technik verwaltet, übertragen, gespeichert werden, spielt Technik eine entscheidende Rolle.
- Genauso wichtig sind z.B. Menschen und ihr Sicherheitsbewusstsein!

Sicherheitsbewusstsein (1)

- Als Bill Clinton das US-Signaturgesetz ratifizierte, unterschrieb er in aller Öffentlichkeit elektronisch mit seinem Passwort „Buddy“.
- Wenigen ist bewusst, dass Email (ohne Zusatzfunktion) keinerlei Sicherheit bietet.
- Passwörter werden unter Tastaturen notiert.
- Passwörter werden zu einfach gewählt.
- Sicherheit wird als Hindernis empfunden.

Sicherheitsbewusstsein (2)

- Eine Vielzahl der möglichen Gefährdungen ist nur mit bewusster oder unbewusster „Unterstützung“ der Mitarbeiter möglich.
- Das Management und die Mitarbeiter müssen die Risiken ihrer Aktionen erkennen können.
- Ein angemessenes Sicherheitsbewusstsein ist die Basis jeder Sicherheitsmaßnahme.
- Wertewandel erforderlich.

Schutzziele

- Informationsintegrität
  - Schutz vor unautorisierter und unbemerkter Modifikation von Daten
- Informationsvertraulichkeit
  - Schutz vor unautorisierter Informationsgewinnung
- Authentizität, Verbindlichkeit
  - Nachweisbarkeit der Herkunft, Schutz vor unzulässigem Abstreiten durchgeführter Handlungen
- Verfügbarkeit
  - Schutz vor unbefugter Beeinträchtigung der Funktionalität von Komponenten

Widersprüche?

- Vertraulichkeit vs. Verfügbarkeit
  - Angenommener Schadensfall: Hacker im System
  - Abwägung zwischen Abschalten des Systems (Vertraulichkeit vor Verfügbarkeit) oder Weiterbetrieb (Verfügbarkeit vor Vertraulichkeit)
- Integrität vs. Vertraulichkeit
  - Dürfen Mitarbeiter verschlüsselte Kanäle aus dem Unternehmen benutzen (SSL, PGP, S/MIME) oder müssen alle Inhalte kontrolliert werden?
  - Wie sind diese Ziele vereinbar?
- Jedes Unternehmen muss eigenen Weg festlegen.
  → IT-Sicherheitsziele und -politik

Schwachstelle

- Verwundbarkeit, Mangel in einem System
- Beispiel Technik
  - E-Mail wurde nur zur Kommunikation im Rechenzentrum entworfen.
- Beispiel Mitarbeiter
  - Notieren sich Passwörter unter Tastatur.
- Führt aus sich selbst heraus noch nicht zu einem Schaden.

Bedrohung

- Umstand, der unter Ausnutzung einer Schwachstelle zu einem Schaden führt.
- Beispiel Technik
  - Abfangen von Emails mit Angeboten durch eine Konkurrenzfirma und Unterbieten
- Beispiel Mitarbeiter
  - Unzufriedener Mitarbeiter manipuliert Personalakten o.ä.

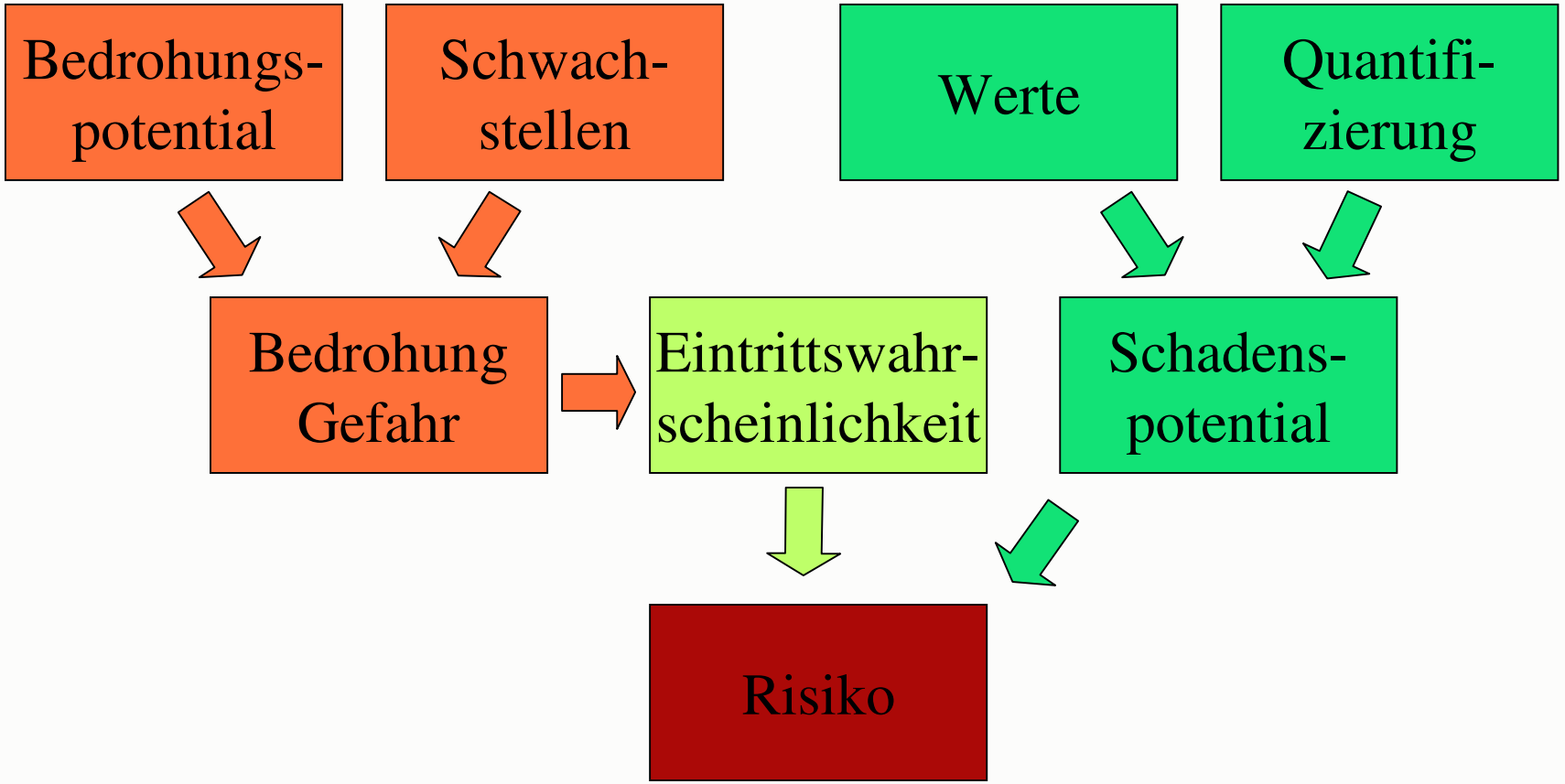
Bedrohungen

- Passive Angriffe, z.B. Abhören von Datenleitungen, Sniffer Attack
- Aktive Angriffe, z.B. Modifizieren, Replay, Spoofing, Denial of Service
- Absichtliches oder unabsichtliches Fehlverhalten
  - z.B. Programmierfehler: Absicht oder Versehen
- Bedrohungen sind nur schwer erkennbar.

Angreifer

- Über 50% aller bekannten Angriffe erfolgen durch Mitarbeiter.
  - Mangelhafte Kenntnisse, Bereicherung, Frust, Fahrlässigkeit
  - Ehemalige Mitarbeiter, deren Accounts nicht gelöscht werden
  - Bekannte Administratorenzugänge
- Hacker, systematische Zerstörung, Spieltrieb
- Wirtschaftsspionage, Geheimdienste
  - Z.B. Echelon-System der National Security Agency (NSA)

Risikoanalyse



Risiko = potentieller Schaden x Eintrittswahrscheinlichkeit

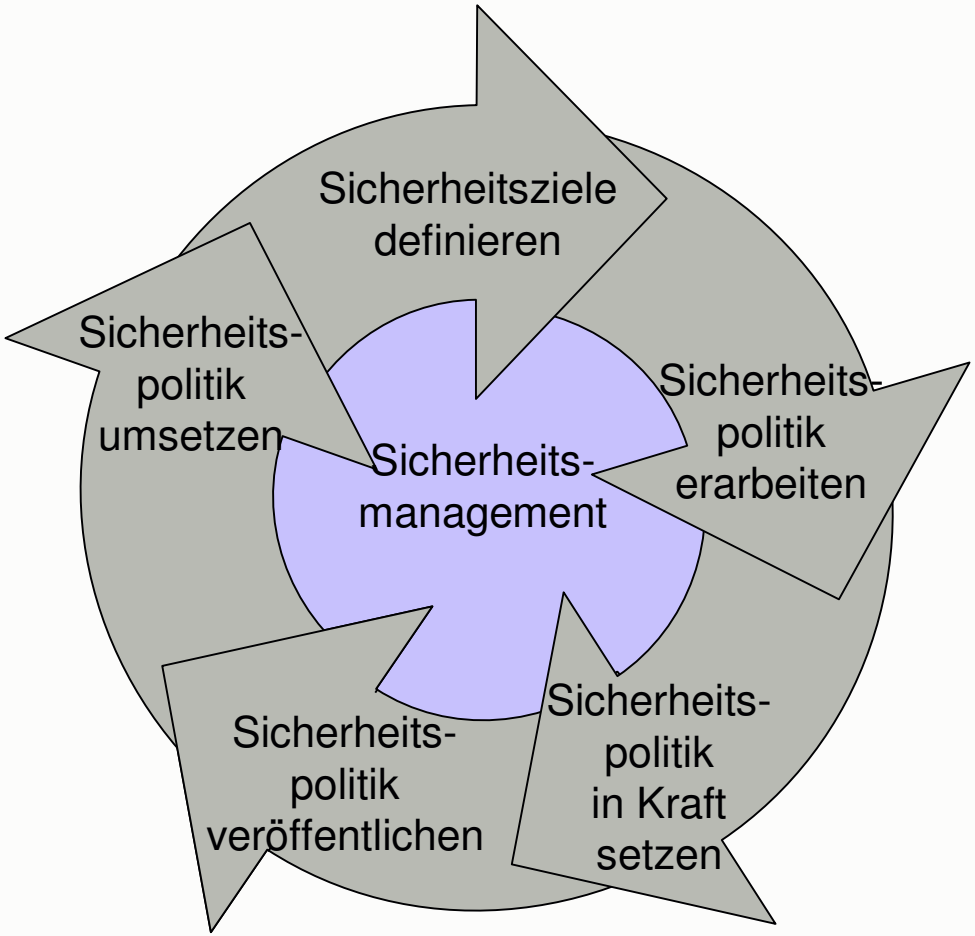
Sicherheitsstrategie

Sicherheitsstrategie (Security Policy)

- Festlegung der Schutzziele
  - Z.B. Vertraulichkeit der Kundendaten
  - Z.B. Verfügbarkeit des Web-Auftritts
- Regeln und Maßnahmen zur Gewährleistung des Erreichens der Schutzziele
  - Rahmenbedingungen (Gesetze, Richtlinien)
  - Organisatorische Maßnahmen (Rollentrennung, Clean Desk)
  - Technische Maßnahmen (Verschlüsselung, Firewalls)

Prozess

- Sicherheitsmanagement muss als **kontinuierlicher Prozess** aufgesetzt werden.
- Jedes Unternehmen muss seine **Sicherheitsziele** definieren.
- Jeder Mitarbeiter muss in seinem Verantwortungsbereich **Sicherheit leben.**
- Dazu werden möglichst **prägnante, einfache Empfehlungen und Anweisungen** benötigt.



Der IT-Sicherheitsprozess

Gesetze, Normen

- Allgemein z.B.
  - KonTraG
    - Gesetz zur Kontrolle und Transparenz im Unternehmensbereich
  - IDW RS FAIT 1,2
    - Institut der Wirtschaftsprüfer
  - BDSG
    - Bundesdatenschutzgesetz
- Bankenspezifisch z.B.
  - Kreditwesengesetz (KWG)
    - MaH, MaK, MaIT (angekündigt)
  - Basel II
    - Nicht nur Kreditrisiken, sondern auch operationelle Risiken

Bsp.: BAFin-Anforderungen

- **Verantwortlichkeit:** Geschäftsleitung
- **IT-Strategie:** Hardware, Softwareauswahl, Sicherheitsanforderungen, Aus- und Weiterbildung
- **Organisation und Funktionstrennung:** Klare Strukturen mit angemessener Funktionstrennung
- **Qualität der Mitarbeiter:** Anforderungen definieren und personalpolitisch umsetzen
- **IT-Reporting:** Berichtspflichten nach klaren Vorgaben
- **Produktionsbetrieb:** Lebenszyklus von Daten, Zugriffsschutzkonzepte, periodische Überprüfung
- **Veränderungsprozesse:** Change-Manangement, Integration und Abnahme
- **Notfallplanung:** Prozesse, Mitarbeiter, Räume, Technik
- **Internes Kontrollsystem:** Gewährleitung der Qualität der Daten und Prozesse, Interne Revision

Die Auswirkungen

- Geschäftsführungsverantwortung
  - Persönliche Haftung
- Aufsichtsrechtliche Fokussierung
- Berichtsrelevant im Jahresabschluss

- Maßnahmen:
  - Sicherheitspolitik
  - Sicherheitsorganisation
  - Risikoanalyse und –management
  - Technik und Organisation

Standards

- Anwenderseitige Sicherheitsaspekte
- ISO/IEC TR 13335 1-5
- BSI IT Grundschutzhandbuch
- BS7799/ISO17799
- IT Infrastructure Library
- NIST Special Publication 800-12
- CoBit
- Canadian Handbook on IT Security
- IT Sicherheitshandbuch für die österreichischen Behörden

- Herstellerseitige IT-Sicherheitsaspekte
- Orange Book
- ITSEC
- CommonCriteria
- FIPS 140

Wie können Standards helfen?

- Best Practice Ansätze liefern Vorgehensweisen zur Policyerstellung, Schwachstellenanalyse, Risikobewertung
  - BS7799 (ISO17799)
    - British Standards Institute (www.bsi-global.com)
    - Etablierung einer Sicherheitspolitik, Zertifizierungsmöglichkeit
  - ISO 13335
    - 5 Reports (Best Practice) zur Umsetzung der Sicherheitsstrategie (www.iso.ch)
  - „Grundschutzhandbuch“
    - Umfangreiche Maßnahmensammlung des deutschen BSI (www.bsi.de)
- Standards liefern Hilfestellung. Umsetzen muss jedes Unternehmen eigenständig.

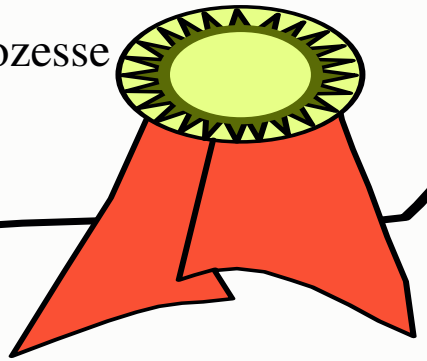
BS7799 / ISO17799

- 2 Bände, herausgegeben vom British Standards Institute (BSI)
- **BS7799-2** formuliert die Anforderungen an ein ISMS (Information Security Management System)
  - Anforderungen
  - Spezifische Maßnahmen im Anhang
  - Das BSI (oder akkreditierte Unternehmen) führt Zertifizierungen nach BS7799-2 durch.
  - Aktuelle Version BS7799-2:2002
- **BS7799-1** dient als Leitfaden (Best Practices) zur Erfüllung der Anforderungen aus BS7799-2
  - Band 1 ist zur internationalen ISO-Norm **ISO17799** erhoben worden.
  - Aktuelle Version ISO17799:2000

BS7799 (ISO 17799)

Maßnahmenkatalog nach BS7799

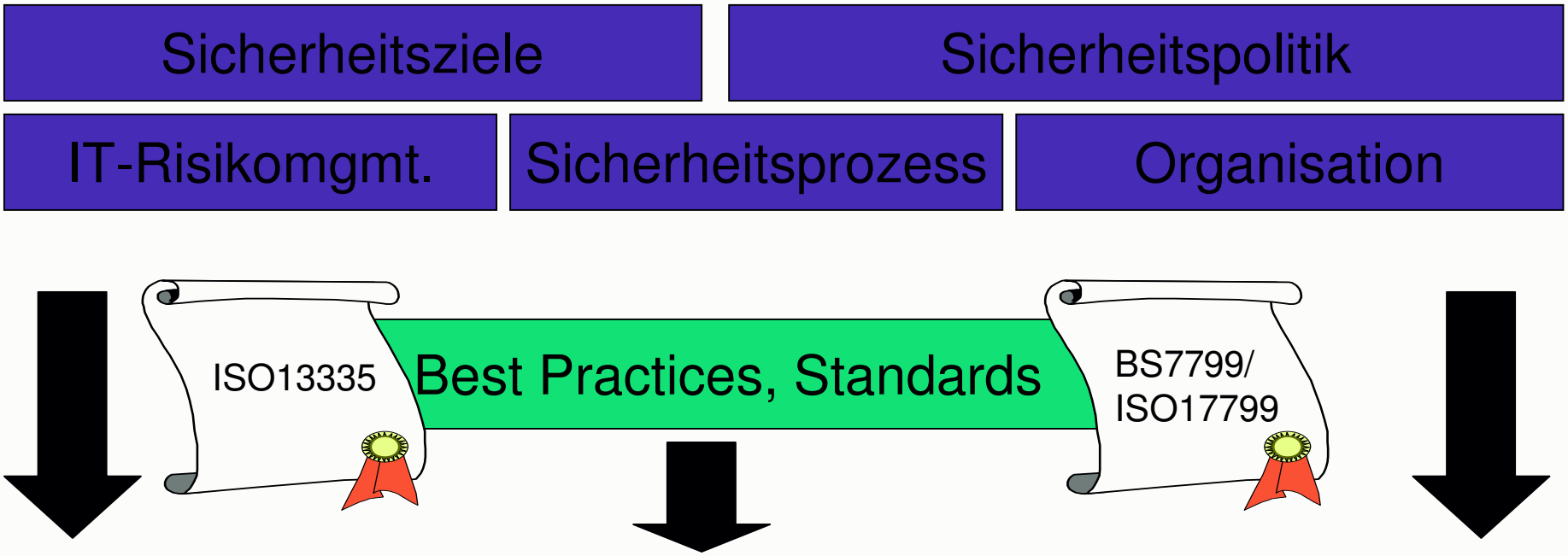
1. Sicherheitspolitik
2. Organisation der Sicherheit
3. Festlegung und Bewertung zu schützender Objekte und Prozesse
4. Physische Sicherheit und Infrastruktur
5. Netzwerk- und Systemmanagement
6. Personelle Sicherheit
7. Zugriffs- und Zugangskontrolle
8. Systementwicklung und -wartung
9. Aufrechterhaltung der Geschäftsprozesse
10. Einhaltung von Verpflichtungen (Compliance)



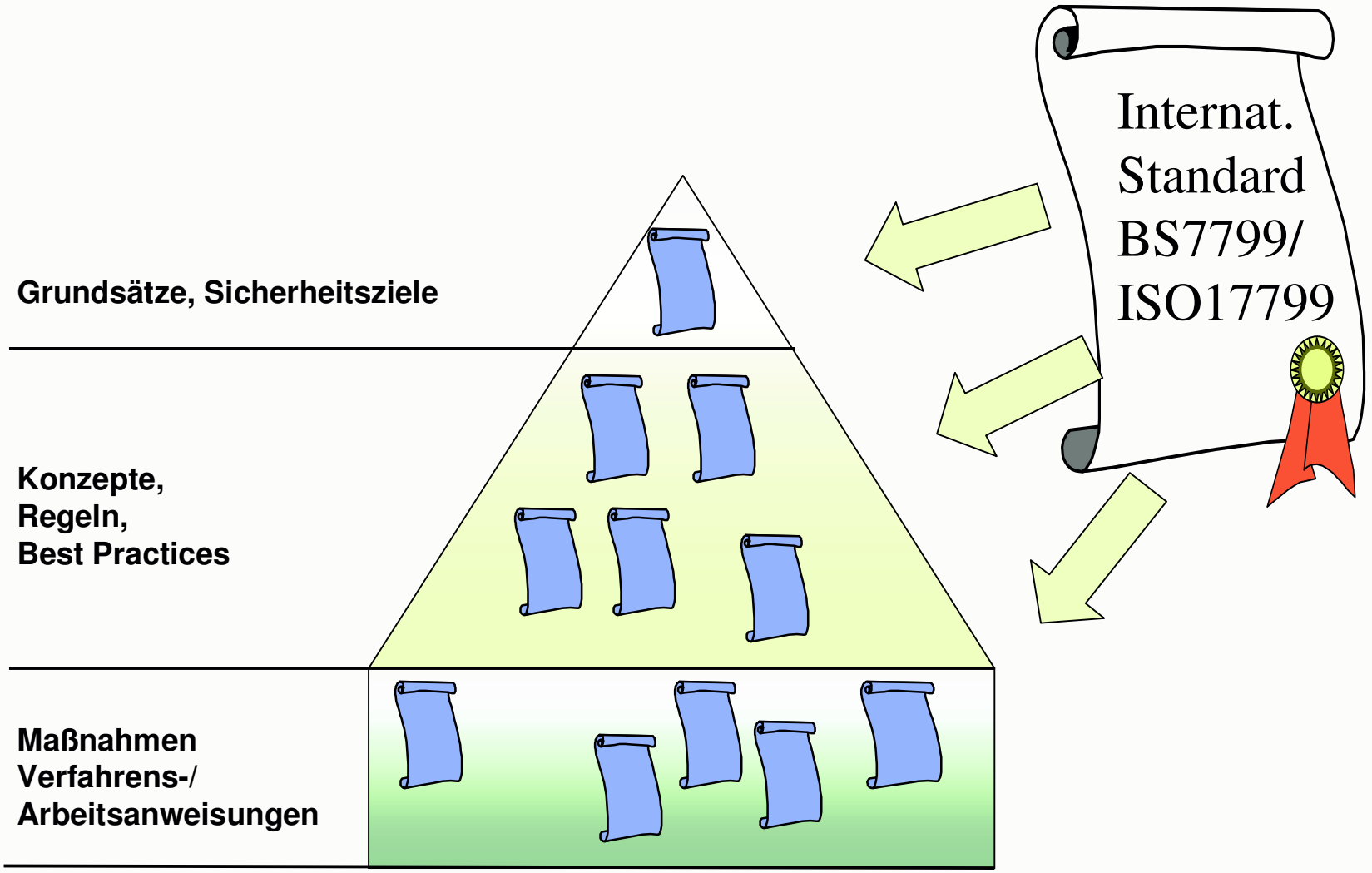
Security nach KISS-Prinzip

- Wer liest heutzutage mehrseitige „Gebrauchsanleitungen“?
- Auch Security verkauft sich nur nach KISS-Prinzip: „Keep it simple, stupid!“
- Sicherheitsarchitekturen und -standards sind etwas für Sicherheitsexperten, Prüfer, Revisoren, aber nicht für Anwender!
- Sicherheitsführer, -checklisten für den Alltag müssen kurz und prägnant sein: „Quick Manuals“ sind in.

Sicherheitsmanagement



Sicherheitsarchitektur



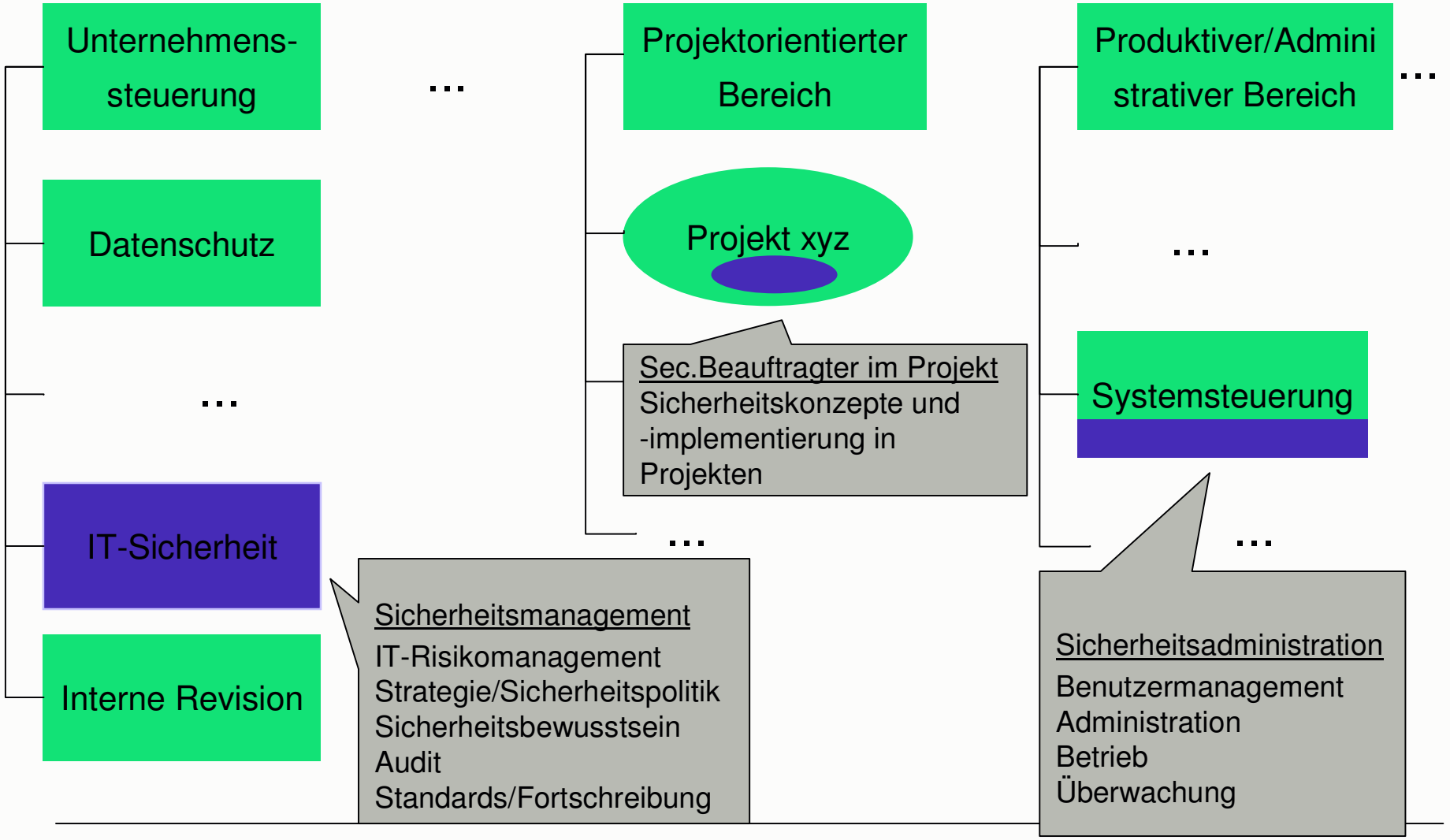
Sicherheitspolitik

- Erarbeitung von Sicherheitszielen anhand der groben Geschäftswerte
  - Einbeziehung aller Unternehmensbereiche
  - Ergebnis: grobe Kategorisierung und Schutzbedarfsbeschreibung der Informationswerte
  - Basis für spätere detailliertere Risikoanalyse
- Verabschiedung einer General IT-Security Policy (die „Verfassung“ für Sicherheit)
  - Inkraftsetzung durch das Management
  - Erlass von Strategien und Grundsätzen zur Erfüllung der Policy, z.B. anhand der in BS7799 enthaltenen 10 Punkte

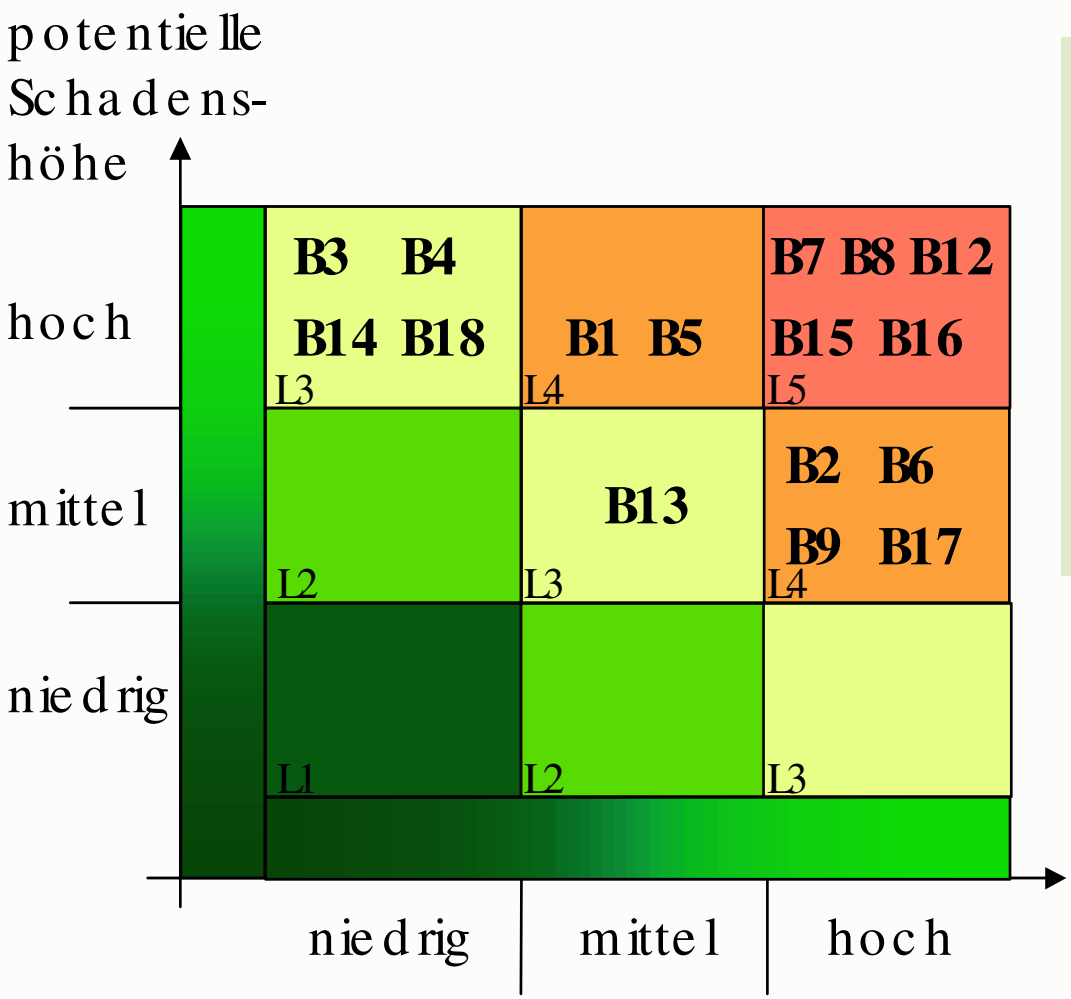
Bestandsaufnahme

- Welche Sicherheitsmaßnahmen sind bereits implementiert?
  - Oft mehr als die Unternehmensleitung weiß!
- Welche davon sind auch (allgemein verfügbar) dokumentiert?
  - Meist weniger als die Unternehmensleitung denkt!
- Gibt es Zielformulierungen des Unternehmens?
  - Aus Unternehmenszielen lassen sich auch Sicherheitsziele ableiten!
- Wie wurden die Mitarbeiter bisher mit Sicherheitsfragen konfrontiert?
- Wer trägt bereits implizite Sicherheitsverantwortung?
- Gibt es bereits ein Risikomanagement für Geschäftsrisiken?
- Wie wurden Ausnahmesituationen bisher behandelt?
- ...

Organisationsbeispiel



Checklisten und Risikoanalyse



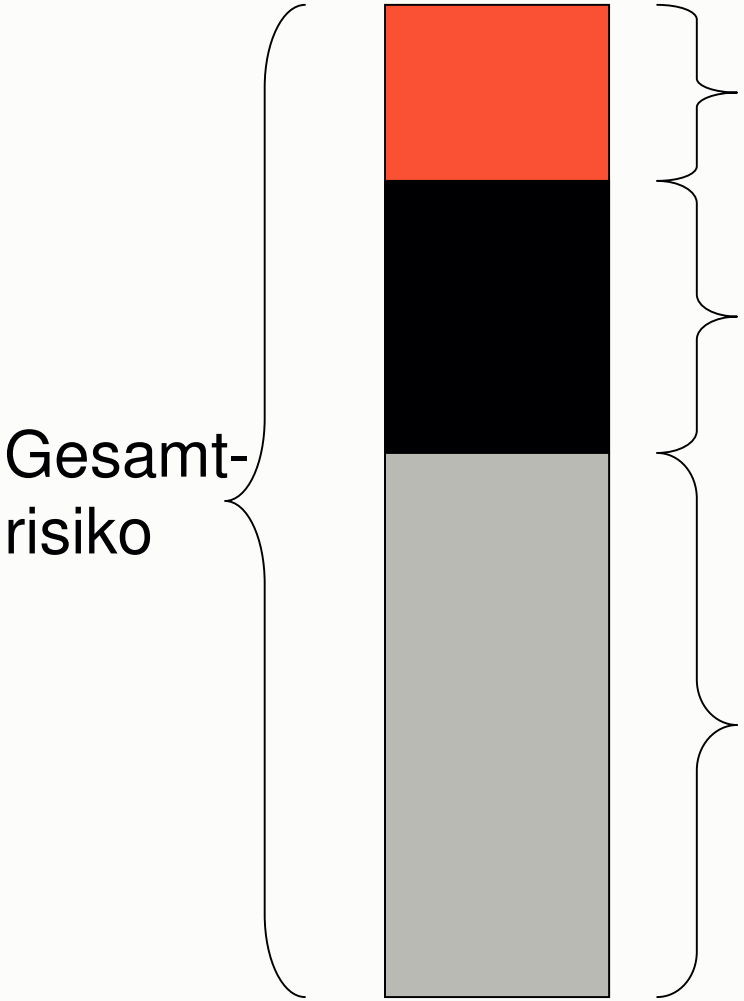
Grundprinzip:

- **Der Risiko-Owner führt die Analyse selbst durch.**

Hilfsmittel:

- **Checklisten,**
- **Bedrohungsszenarien,**
- **Klassifikationsschemata,**
- **Beratung durch die Sicherheitsabteilung.**

IT-Risikomanagement



Akzeptiertes Restrisiko

- Befristete Risikoübernahme

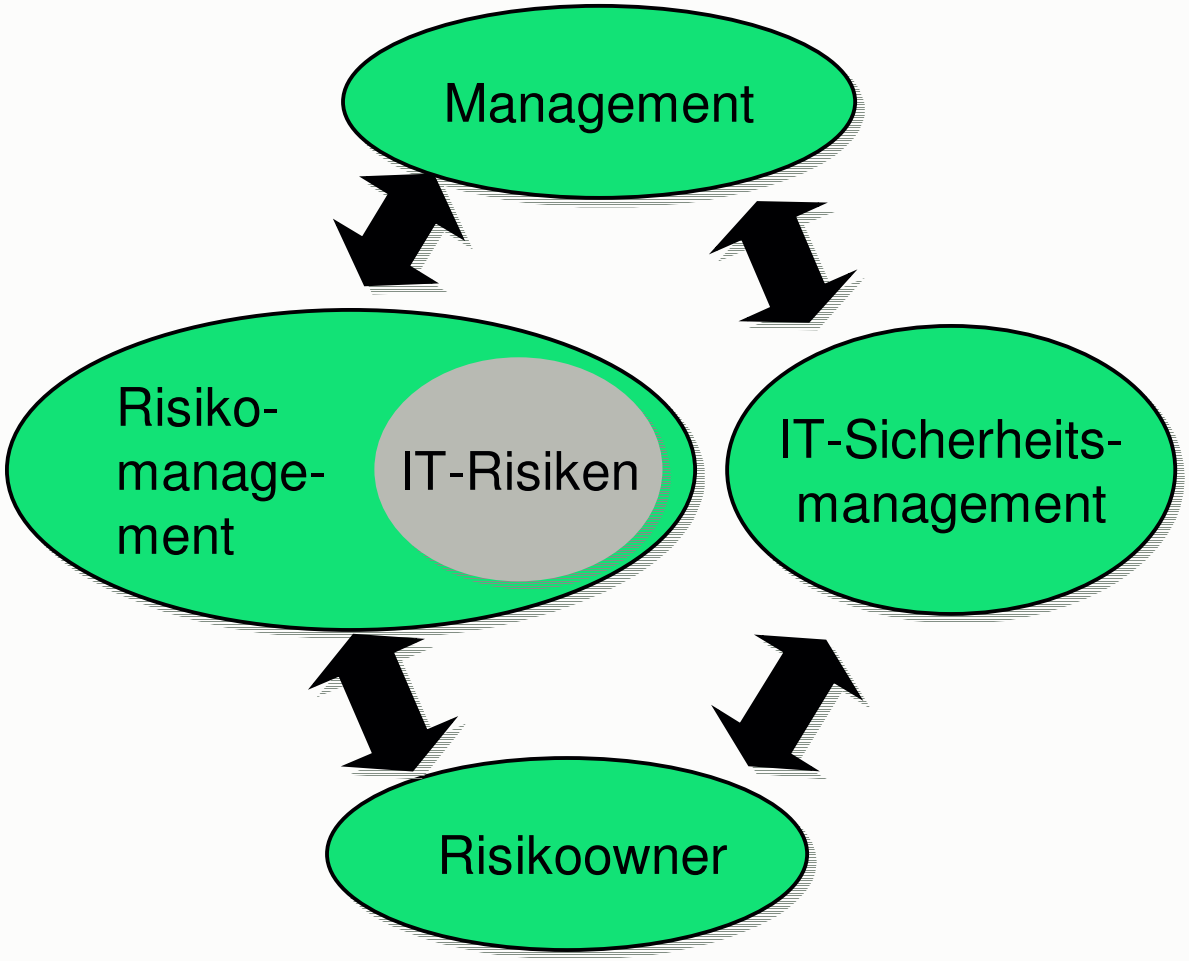
Organisatorische Maßnahmen

- Verfahrens- und Arbeitsanweisungen
- Vieraugenprinzip

Technische Maßnahmen

- Virenschutz
- Firewalls
- Verschlüsselung
- Rechteverwaltung
- IDS

Gesamtrisikomanagement



- IT-Risiken sind in **ein** Teil der Unternehmensrisiken.
- Das IT-Sicherheits-management unterstützt bei der Feststellung der IT-Risiken.
- Eine Einordnung in die Gesamtrisiken führt zu einer objektiveren Bewertung der IT-Risiken durch das Management.
- Nicht sofort zu beseitigende Risiken müssen durch das Management per Unterschrift (zeitlich befristet) übernommen werden.
- Das Management kann Sicherheit nicht mehr „wegdelegieren“.

Sicherheitsmanagement ...

- stellt teure und hohe Sicherheitsanforderungen,
- leitet punktuelle Sicherheitsmaßnahmen ein,
- beschränkt sich auf Sicherheitstechnik,
- stellt ein starres Regelwerk dar,
- geht nur die Abteilung Sicherheit an,
- reagiert nur auf offensichtliche Schwachstellen,
- behindert den Wertschöpfungsprozess.

- bewertet Bedrohungen und Risiken im Gesamtkontext,
- leitet technische, organisatorische oder andere Maßnahmen zur Risikominimierung ein und berät bei deren Umsetzung,
- passt sich Veränderungen im Unternehmen an und ist ein kontinuierlicher Prozess,
- betrifft jeden Mitarbeiter,
- sorgt für die Einhaltung eines stabilen Sicherheitsniveaus sowie gesetzlicher und sonstiger Anforderungen,
- ist ein Business-Enabler.

Einige Bedrohungen

- Kapitel 2: Einige Bedrohungen
  - Viren, Würmer, Trojaner
  - Buffer Overflow
  - Angriffe auf Web-Anwendungen

Viren, Würmer, Trojaner

- Allgemeiner Begriff: Malware (**Mal**icious Soft**ware**)
- Programme mit beabsichtigter, bösartiger Funktionalität
- Augenfälligste Ursache für Sicherheitsprobleme
- Häufigste?
- Erhebliche wirtschaftliche Schäden

Chronik

- 1984 Fred Cohen: „Computer Viruses – Theory and Experiments“
- 1986 Erste PC-Viren
- 1991 Bootvirus „Form“
- 1991 Michelangelo – Schaden vor allem durch Panik
- 1995 Erster Makrovirus „Concept“ – infiziert Word-Dokumente
- 2003 MSBlaster, Sobig, ...

Vorfälle

Number of incidents reported (CERT/CC)

1988-1989

Year	1988	1989
Incidents	6	132

1990-1999

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Incidents	252	406	773	1,334	2,340	2,412	2,573	2,134	3,734	9,859

2000-2003

Year	2000	2001	2002	1Q-3Q 2003
Incidents	21,756	52,658	82,094	114,855

Virus

- Nicht eigenständig lauffähig
- Wirtsprogramm nötig
- Reproduzierfähig
- Infektion
- Enthalten oft Schadfunktion
- Verbreitung (Quelle: ICSA 1999)
  - Email (>50%), Download
  - Makroviren in Word-, Excel-Dokumenten (ca. 70%)
- Verbreitungswege über das Internet
  - Email, Web-Download

Virus

- Kategorien
  - Dateiviren
  - Bootsektorviren
  - Makroviren
- Virus Construction Kits

Wurm

- Eigenständig lauffähiges Programm
- Reproduktionsfähig
- Enthalten oft Schadfunktion
- Verbreitung
  - Ohne menschliches Zutun
  - Über das Netzwerk
  - Häufig über Buffer-Overflow-Angriffe auf Systemprozesse
  - Oder über autonomen Email-Versand
- In großen Netzen (Internet, Firmennetz) hoher Beseitigungsaufwand

Die Gründe

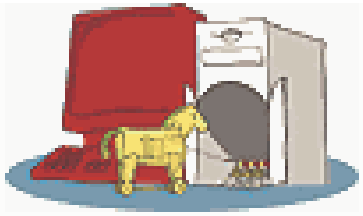
- Betriebssystemmonokultur (Microsoft)
- Zunehmende Internet-Vernetzung
- Kein Spezialwissen mehr erforderlich (Construction Kits, Script Kiddies, ...)

Die Schäden

- Verfügbarkeitsverluste
- Beseitigungsaufwand
- Vorsorgeaufwand
  - CERT-Meldungen
  - Einspielen von Patches
  - Anti-Viren-Programme
  - Aktualität von Viren-Pattern

Wurm-Chronik

- 1988 Internet-Wurm: 6000 Infektionen
- 1999 Melissa:100000 Infektionen
- 2000 ILOVEYOU
- 2003 MSBlaster bzw. Lovesan



Trojaner

- Programm, das neben den offiziellen weitere absichtlich hinterlegte, verborgene Funktionalitäten enthält.
- Dient der Ausspähung, Veränderung von Daten oder der Erlangung von Rechten etc.
- Übertragung z.B. per Virus, Wurm oder Download
- Gegenmaßnahmen z.B.:
  - Einschränkung von Rechten auf das Notwendige,
  - Speicherung sensibler Daten nur in verschlüsselter Form.

Aktive Inhalte

- Mobiler Code, der evtl. aus nicht vertrauenswürdiger Quelle stammt.
- Beispiel: ActiveX, Java Applets, JavaScript
- Angriffe auf mobilen Code sind möglich.
- Angriffe mittels mobilem Code auf Gastrechner:
  - Ausspähung, Veränderung von Daten
  - Z.B. Auslesen der Passwortdatei und Übermitteln zum bequemen Ansetzen von Crackprogrammen
  - Z.B. Ausspähen der TAN-Liste

Gegenmaßnahmen

- Schutz des Codes:
  - Verschlüsselung bei Übertragung
  - Jedoch: Schutz bei der Ausführung nicht möglich
- Schutz des Rechnerumgebung:
  - Isolierung der Ausführungsumgebung (z.B. Java Sandbox)
  - Digitale Signatur durch den Erzeuger (z.B. Microsoft Authenticode)
    - Aber: Digitale Signaturen lassen sich in diesem Fall billig und einfach erzeugen.
    - Eine Signatur beweist keine Harmlosigkeit, sondern höchstens die Herkunft.
    - Verantwortung über Ausführung bleibt beim Benutzer, der die Funktionalität will und Warnungen wegklickt.

Buffer Overflow

- Wichtigste Klasse von Software-Schwachstellen
- Firewall bietet keinen Schutz bei Software-Schwachstelle in Netzwerkdienst oder –anwendung.
- Bedrohungen:
  - Denial-of-Service
  - Kompromittierung der Anwendung oder des Rechners über das Netz (fatal)

Buffer Overflow

- Verursacht durch Programmierfehler.
  - Fehlende oder falsche Überprüfung der Länge von Benutzereingaben und
  - Speicherung der Eingabe in Puffer mit statischer Anzahl von Elementen.
- Die Programmiersprachen C und C++ überprüfen nicht die Einhaltung von Puffergrenzen bei Array-/Zeiger-Referenzierungen.
  - Zudem Bibliotheksfunktionen (libc, WinAPI) teilweise „unsicher“.
- Programmierer oft zu faul oder unkundig.

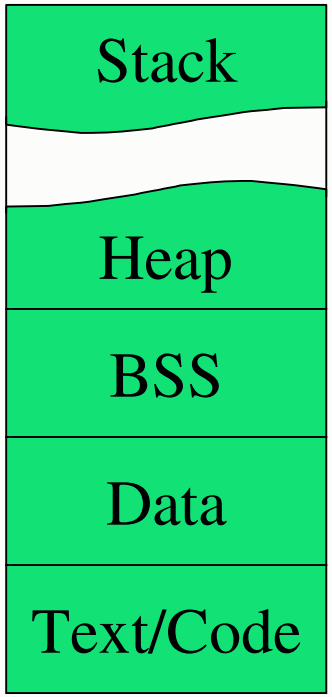
Buffer Overflow

- Beispiel:

```
int
main (int argc, char *argv[])
{
  if (argc > 1)
  {
     char buff [512];
     strcpy (buff, argv[1]);
  };
  return 0;
}
```

Buffer Overflow

- Wir beschreiben hier nur grob stack-basierte Buffer-Overflows.
- Prozessspeicher



Hohe Adressen

Niedrige Adressen

Buffer Overflow

- Bei IA-32-Architektur wächst der
  - Stack in Richtung niedrigerer Adressen,
  - Puffer in Richtung höherer Adressen.
- Stack kann gezielt durch Buffer-Overflow überschrieben werden.
- Stack Frame einer Funktion enthält u.a. die Rücksprungadresse der aufrufenden Funktion.
- Manipulation der Rücksprungadresse ermöglicht:
  - Denial-of-Service
  - Ausführung von Code (aus Text-Bereich oder Bibliothek bzw. eingeschleust)

Buffer Overflow

- In der Regel wird versucht Shell-Code einzuschleusen.
- Shell läuft mit den Rechten des kompromittierten Prozesses, ggf. Root-Shell!
- Ausnutzen einer Buffer-Overflow-Schwachstelle ist nicht so schwer, wie man (vielleicht) denkt, wenn sie erst einmal gefunden wurde.

Schutzmaßnahmen

- Bekämpfung der Ursachen (**Software-Entwicklung**)
  - Schulung von Programmierern
  - Programmierrichtlinien
  - Berücksichtigung der Sicherheit im Entwicklungsprozess
  - Wahl der Programmiersprache
  - Software-Tests und –Audits
  - ...
  - Nur: I.d.R. wird Kauf-Software eingesetzt.

Schutzmaßnahmen

- Einspielen von Security Patches (**reaktiv**)
  - Zwingend erforderlich.
  - Zeitspanne
    - zwischen Entdeckung/Bekanntwerden der Schwachstelle und Verfügbarkeit des Patches
    - zwischen Verfügbarkeit des Patches und Einspielen des Patches („Never touch a running system“; Priorität anderer Aufgaben des Administrators)
  - Nicht zu unterschätzender Aufwand für die Administratoren.
  - Patches liefern oft weitere Schwachstellen mit.

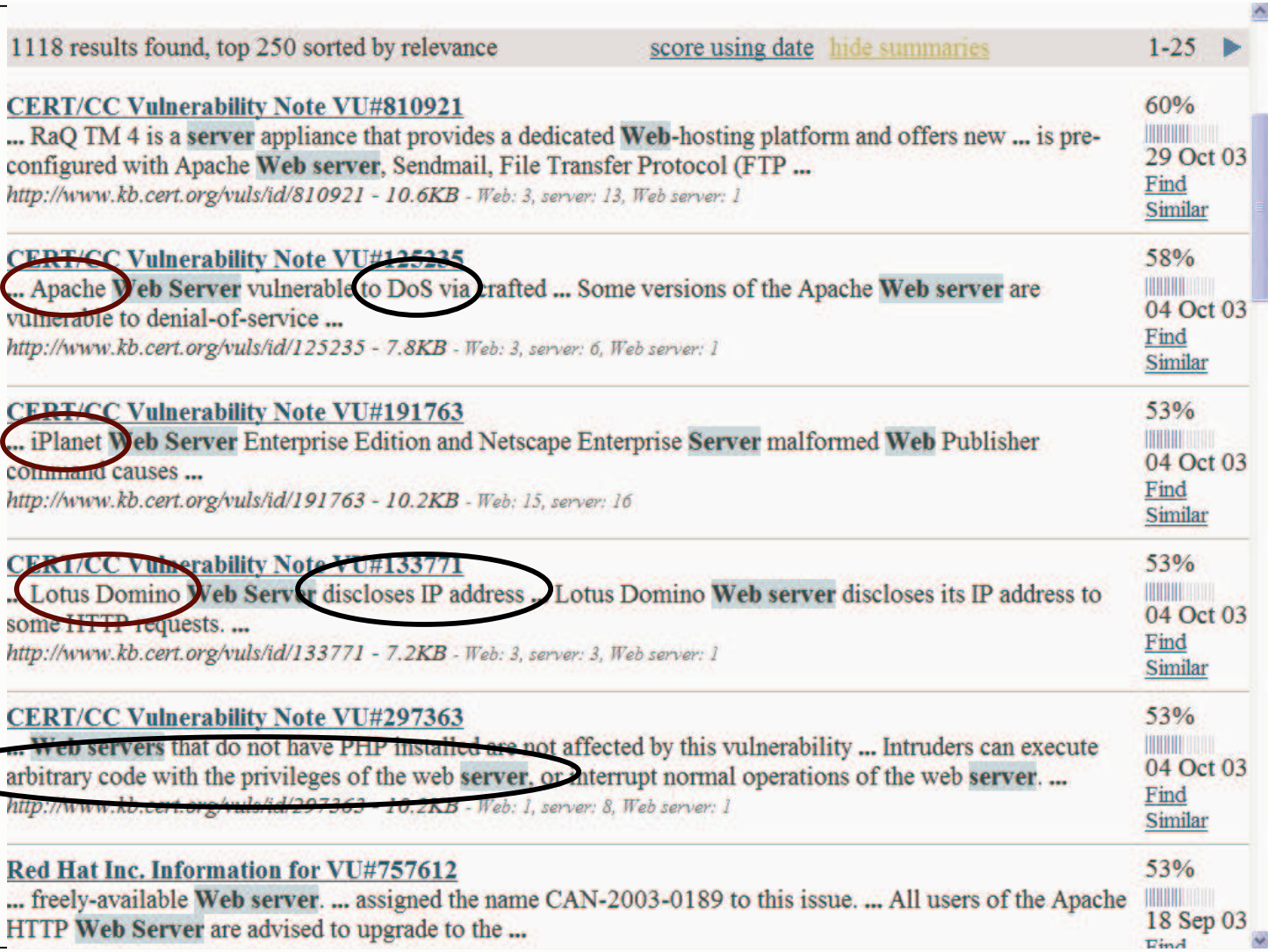
Schutzmaßnahmen

- Eindämmung der Auswirkungen bei Kompromittierung (**proaktiv**)
  - Härtung des Systems (Berechtigungen, gestartete Dienste, ...)
  - Trusted Operating Systems, Intrusion Prevention Systems, ...
  - „Sichere“ Prozessumgebungen und Bibliotheksfunktionen
  - ...
  - Systeme werden komplexer.
  - Betriebsaufwände und –probleme können steigen.

Angriffe auf Web-Anwendungen

- an einer Vielzahl von Stellen mit unterschiedlichen Erfolgsaussichten und Schadenspotenzialen möglich.
  - Client-Rechner (z.B. Trojaner)
  - Angriffe auf die Kommunikationssicherheit (z.B. Man-in-the-middle)
  - Web-Server
  - Betriebssystem
  - Anwendung (z.B. Parameter Manipulation, SQL Injection, Cookie Poisoning)
  - Interne Datenbank
- Eine Firewall allein bietet hier keinen ausreichenden Schutz.
- HTTP-Zugriff reicht!

Web-Server Schwachstellen



1118 results found, top 250 sorted by relevance — score using date — hide summaries — 1-25

CERT/CC Vulnerability Note VU#810921 — 60%
... RaQ TM 4 is a **server** appliance that provides a dedicated **Web**-hosting platform and offers new ... is pre-configured with Apache **Web server**, Sendmail, File Transfer Protocol (FTP ...
http://www.kb.cert.org/vuls/id/810921 - 10.6KB - Web: 3, server: 13, Web server: 1
29 Oct 03 — Find Similar

CERT/CC Vulnerability Note VU#125235 — 58%
... Apache **Web Server** vulnerable to DoS via crafted ... Some versions of the Apache **Web server** are vulnerable to denial-of-service ...
http://www.kb.cert.org/vuls/id/125235 - 7.8KB - Web: 3, server: 6, Web server: 1
04 Oct 03 — Find Similar

CERT/CC Vulnerability Note VU#191763 — 53%
... iPlanet **Web Server** Enterprise Edition and Netscape Enterprise **Server** malformed **Web** Publisher command causes ...
http://www.kb.cert.org/vuls/id/191763 - 10.2KB - Web: 15, server: 16
04 Oct 03 — Find Similar

CERT/CC Vulnerability Note VU#133771 — 53%
... Lotus Domino **Web Server** discloses IP address ... Lotus Domino **Web server** discloses its IP address to some HTTP requests. ...
http://www.kb.cert.org/vuls/id/133771 - 7.2KB - Web: 3, server: 3, Web server: 1
04 Oct 03 — Find Similar

CERT/CC Vulnerability Note VU#297363 — 53%
... **Web servers** that do not have PHP installed are not affected by this vulnerability ... Intruders can execute arbitrary code with the privileges of the web **server**, or interrupt normal operations of the web **server**. ...
http://www.kb.cert.org/vuls/id/297363 - 10.2KB - Web: 1, server: 8, Web server: 1
04 Oct 03 — Find Similar

Red Hat Inc. Information for VU#757612 — 53%
... freely-available **Web server**. ... assigned the name CAN-2003-0189 to this issue. ... All users of the Apache HTTP **Web Server** are advised to upgrade to the ...
18 Sep 03 — Find

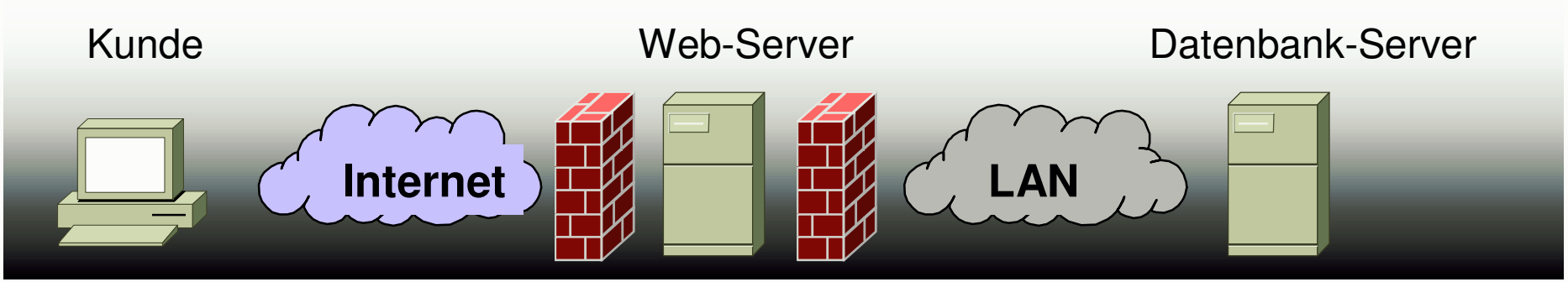
Web-Server Schwachstellen

- Suche nach „Web-Server“ in CERT/CC Vulnerability Notes zeigt:
- Bei jedem Web-Server werden regelmäßig Schwachstellen mit teils fatalen Auswirkungen gefunden:
  - Denial-of-Service
  - Ausführung beliebigen Codes mit den Rechten des Web-Servers (oft Root)
  - Ungewollte Veröffentlichung von internen Informationen (Bsp.: Unicode Exploit)
  - ...

Unicode Exploit

- Microsoft's Information Server (IIS) unterstützt Unicode-Kodierungen in URLs.
- Beispiel („hex-encode“ Schema):
  `http://sicherer.server.bogus/cgi-bin/..%c0%af..%c0%af../winnt/system32/cmd.exe`
- `..%c0%af..%c0%af..` dekodiert: `../../..`
- Schwachstelle: Pfadprüfung für Zugriffskontrolle wurde vor der Dekodierung ausgeführt.
- Resultat: Ausführung von
  `C:\IIS\InetPub\cgi-bin/../../../winnt/system32/cmd.exe`

SQL Injection



Kunde meldet sich mit Username und Passwort an.

Eingabe

Anmelde-Seite

Username: anton

Passwort: ******

Datenbank-befehl

"select * from users where username = '" + username + " ' and password = '" + password + "'"

Hacker gibt präparierte Strings ein.

Eingabe

Username: ' or 1=1--

Passwort:

Datenbank-befehl

Wirkung?

SQL Injection

SQL-Abfragestring

- "select * from users where username = '" + username + " ' and password = '" + password + "'"

Übung

- Welche Wirkung haben die folgenden Eingaben?
- Username: '; drop table users--
  Password:
- Username: admin'--
- Username: ' or 1=1--

Parameter-Manipulation

- Da HTTP zustandslos ist, werden Informationen von der Anwendung oft client-seitig abgelegt:
  - hidden field
  - Cookie
  - vorbereitete URL
- Sind alle vom Anwender manipulierbar.
- Programmierer nehmen oft auch sicherheitsrelevante Daten ohne Prüfung entgegen.

Beispiele

- Preisänderung bei Shopping-Anwendung
  - <type=hidden name=preis value=99.95>
- Zugriff auf Systemdateien
  - Suche.cgi?template=result.html durch /etc/passwd ersetzen

Kryptographie

- Kapitel 3: Kryptographie
  - Symmetrische Kryptographie
  - Hash Algorithmen, Message Digests
  - Public Key Kryptographie
  - Schlüssel-Management

Kryptographische Verfahren

- **Kryptographie**
  - Wissenschaft, die sich mit der Absicherung von Nachrichten beschäftigt.
- **Kryptoanalyse**
  - Kunst, Chiffretext aufzubrechen.
- **Kryptologie**
  - Zweig der Mathematik, der Kryptographie und Kryptoanalyse umfasst.
- **Steganographie**
  - Methode zum Verbergen der Existenz einer Nachricht.

Griechisch: steganos = verdeckt; kryptos = geheim; graphein = schreiben

Ziele der Kryptographie

- **Vertraulichkeit** (Confidentiality)
- **Datenintegrität** (Data integrity)
- **Authentifizierung** (Authentication) bzw. Authentizität
- **Verbindlichkeit** (Non-repudiation)

Bewertungskriterien:

Sicherheitsniveau, Funktionalität, Betriebsmethoden, Performanz, Leichtigkeit der Implementierung

Kryptographisches System

1. Menge von Klartextnachrichten M,
2. Menge von Chiffretextnachrichten C,
3. nicht-leere Menge von Verschlüsselungs-Schlüsseln EK,
4. nicht-leere Menge von Entschlüsselungs-Schlüsseln DK

   Bijektion $f : EK \rightarrow DK$, $f(k_E) = k_D$

Kryptographisches System

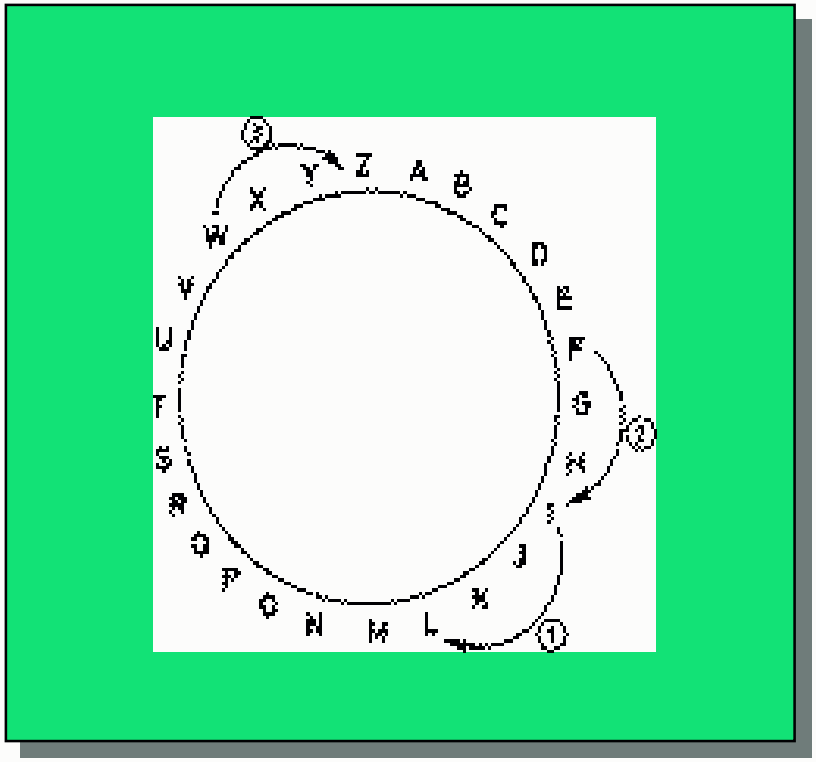
5. Verschlüsselungsverfahren
   $E : M \times EK \rightarrow C$, $E(m,k) = c$
   wobei $E_k : M \rightarrow C$ injektiv für jedes $k \in EK$
   (auch $E_k(m)$ statt $E(m,k)$)
6. und Entschlüsselungsverfahren
   $D : C \times DK \rightarrow M$, $D(c,k) = m$ mit
   $D(E(m,k_E),k_D)) = m$ mit $f(k_E) = k_D$
   (auch $D_k(c)$ statt $D(c,k)$)

Beispiel: **Caesar-Chiffrierung**
Verschieben der Buchstaben um
k Positionen nach rechts

Verschlüsselungsverfahren

Caesar-Chiffre:

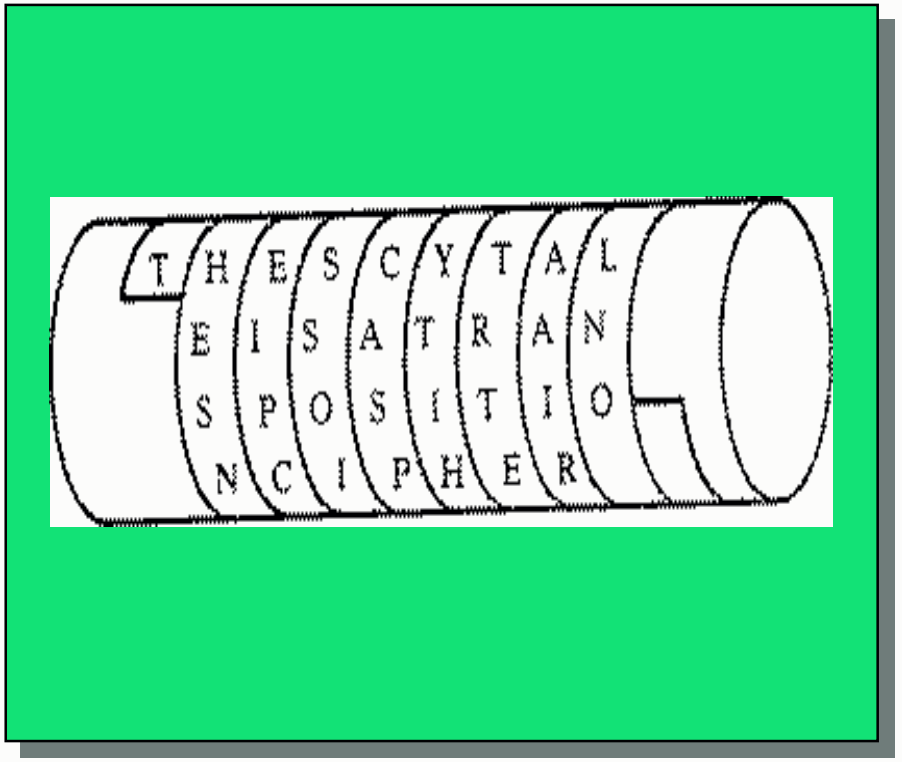
Prinzip der Substitution



Scytale Chiffre:

Prinzip der Transposition

oder Permutation



Kryptographische Systeme

- Symmetrische Verfahren (secret-key)
  - $k_E = k_D$, geheimer Schlüssel
  - Beispiele: DES (Data Encryption Standard), Blowfish, IDEA, RC2, RC4, RC5, AES (Advanced Encryption Standard, Rijndael)
- Asymmetrische Verfahren (public-key)
  - Öffentlicher (public) und privater Schlüssel (private key)
  - Bekanntestes Verfahren: RSA

Sicherheit

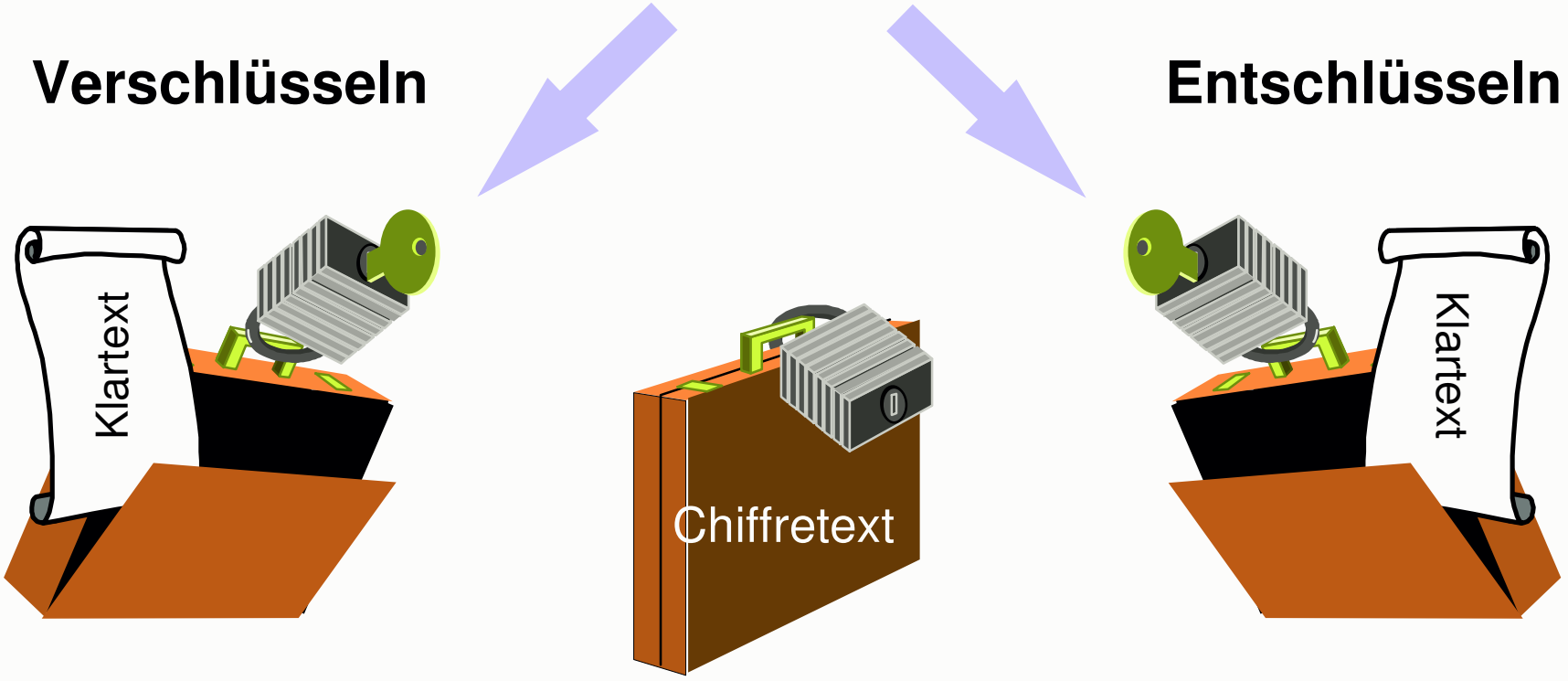
- Sicherheit darf nicht von **Geheimhaltung** der Ver- und Entschlüsselungsfunktion abhängen.
- Verschlüsselung darf ohne Kenntnis des Schlüssels nicht in einer angemessenen Zeit – abhängig von der Lebenszeit der geschützten Daten – **aufzubrechen** sein.
- Exhaustive Search: **Schlüsselraum** EK muss sehr groß sein. (Notwendig, aber nicht ausreichend!)
  - *Beispiel:* 56-Bit Schlüssel (z.B. DES): Schlüsselraum = 2^{56} unterschiedliche Schlüssel
- Anforderung an **Schlüssellänge**
  - symmetrische Schlüssellänge >= 128 Bit
  - RSA-Schlüssellänge >= 1024 Bit

Symmetrische Kryptosysteme

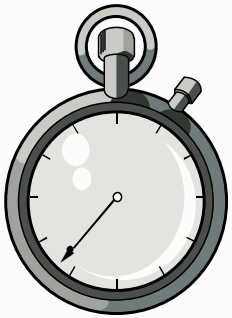
Identischer geheimer Schlüssel (secret key)

zum

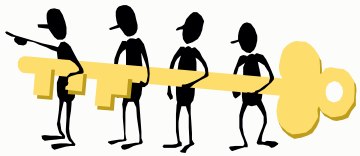
Verschlüsseln **Entschlüsseln**



Symmetrische Kryptosysteme



- **Effizient in Hard- und Software zu implementieren**
  z.B. DES, 3DES, NEU: RIJNDAEL (AES)



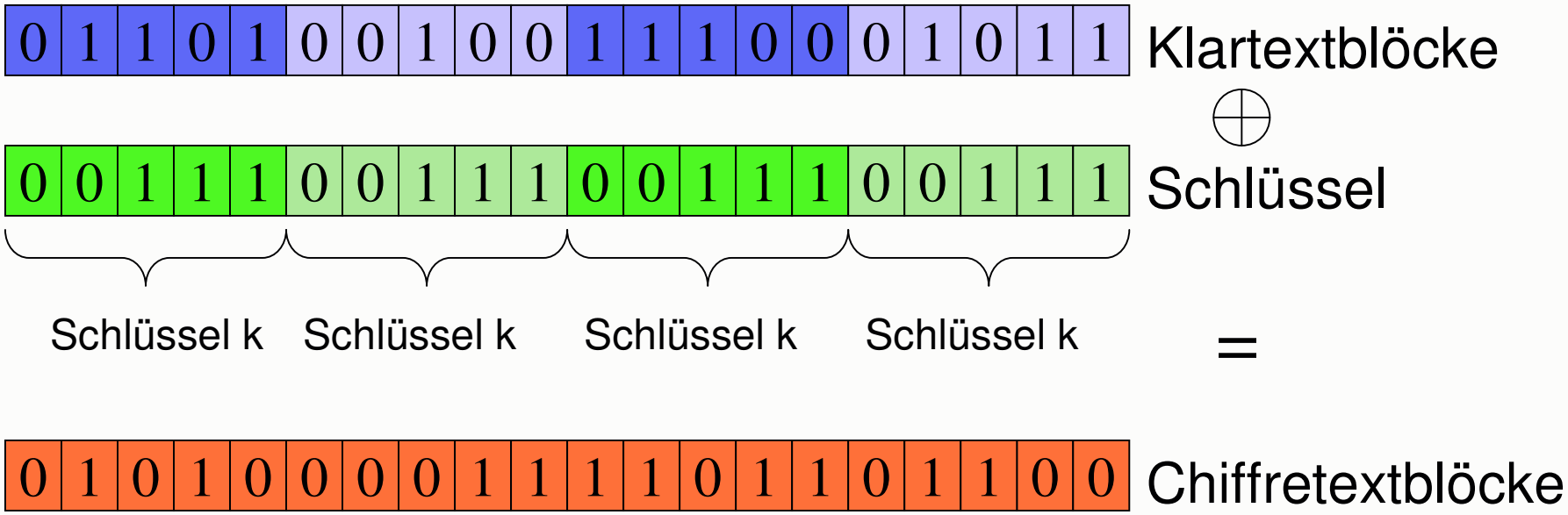
- **Problem der geheimen Schlüsselverteilung**
  n Partner benötigen n*(n-1)/2 Schlüssel

- **Keine Nachweisbarkeit der Herkunft einem Dritten gegenüber**

Symmetrische Kryptosysteme

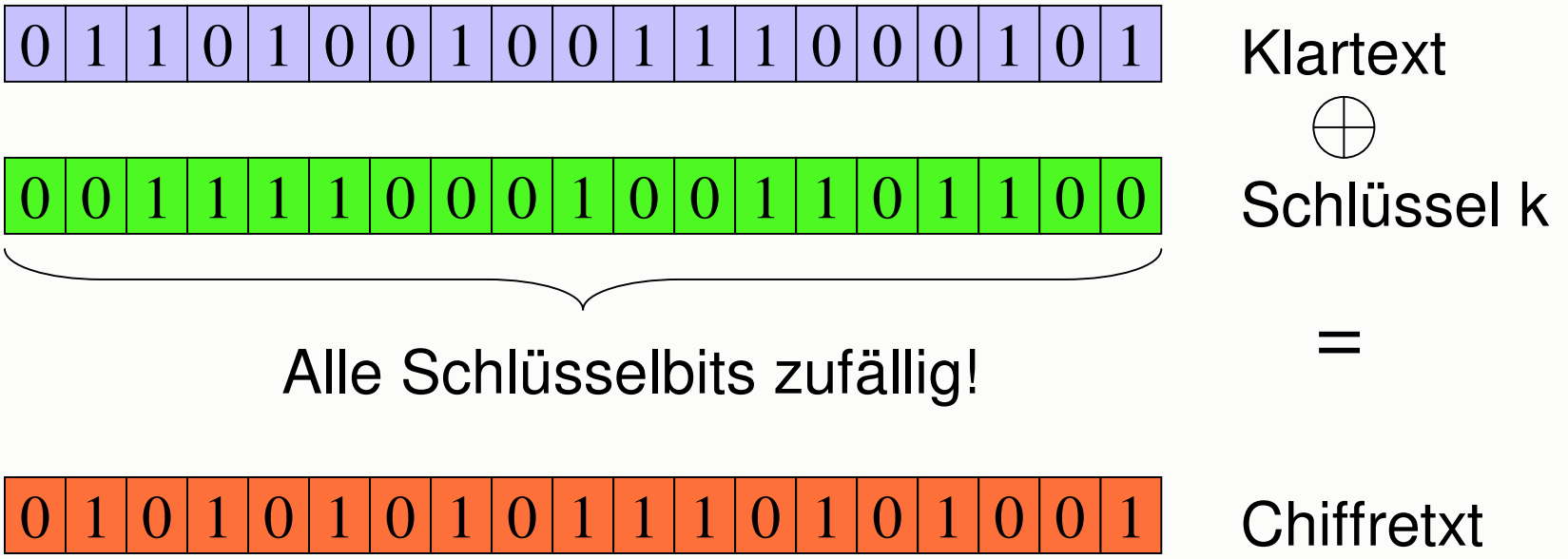
- 2 Klassen symmetrischer Verfahren
  - **Blockchiffre**: Blöcke (Strings) fester Länge; jeder Block mit gleicher Funktion verschlüsselt.
  - **Stromchiffre**: (Kleine) Einheiten mit Schlüsselstrom verschlüsselt.
- One-Time Pad
  - Schlüssel gleich lang wie Klartext, zufällig und niemals wiederverwendet.
    → absolut sicher!

Blockchiffre



Verschlüsseln von Blöcken fester Länge.
Evtl. Auffüllen (Padding) des Klartextes am Ende nötig.

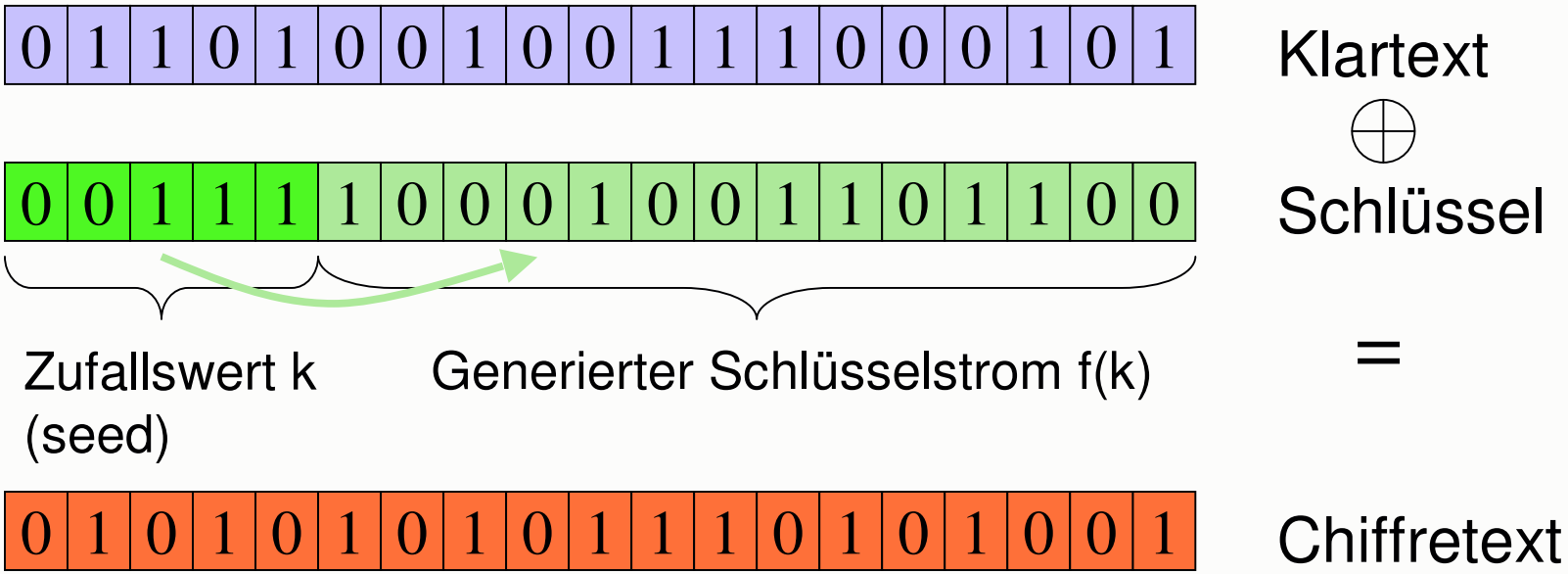
One-Time Pad



Perfekte Sicherheit: Falls der Schlüssel einmalig, zufällig gewählt und genauso lang wie der Klartext ist, ist der Chiffretext ebenfalls zufällig und daher beweisbar sicher.

→ **Absolute Unpraktikabilität**

Stromchiffre



Idee: Generierung einer pseudozufälligen (= zufällig aussehenden) Schlüsselsequenz aus einem kurzen Anfangswert. In f kann z.B. auch der vorher schon generierte Chiffretext eingehen.

Blockchiffren

- Klassische Verschlüsselungstechniken:
  - **Transposition**, Permutation: Vertauschen der Anordnung der Klartextzeichen
  - **Substitution**: Ersetzen von Zeichen (z.B. Caesar-Chiffre)
- **Produktchiffre** (z.B. DES)
  - Verknüpfungen aus Transposition und Substitution (Runden)
- Transposition fügt **Diffusion** hinzu, Substitution fügt **Konfusion** hinzu.
- Ver- und Entschlüsselung basiert auf einfachen Operationen (u.a. Shifts, XOR).
  ➔ Effizient in Hard- und Software zu implementieren.

Modi von Blockchiffren

- **Electronic Code Book** (ECB)
  - ein Klartextblock in einen Chiffretextblock verschlüsselt → Problem: Block Replay: Blöcke entfernen, wiederholen oder austauschen
- Chaining bewirkt Rückkopplung.
- **Cipher Block Chaining** (CBC)
  - XOR-Verknüpfung mit vorherigem Chiffretextblock
  - identische Anfänge → Initialisierungsvektor
- **Output Feedback** (OFB)
- **Cipher Feedback** (CFB)
- Kriterien: Sicherheitsprobleme, Fehlerfortpflanzung, Synchronisierung von Stromchiffrierungen

Data Encryption Standard

- Seit 1977 genormt, bis 1998 Standard.
- Große Akzeptanz und Verbreitung: u.a. Banken-Umfeld, DES-Chips
- 1998 lief die letzte Zertifizierungsperiode des DES aus.
- 1997: Ausschreibung für den AES (Advanced Encryption Standard)

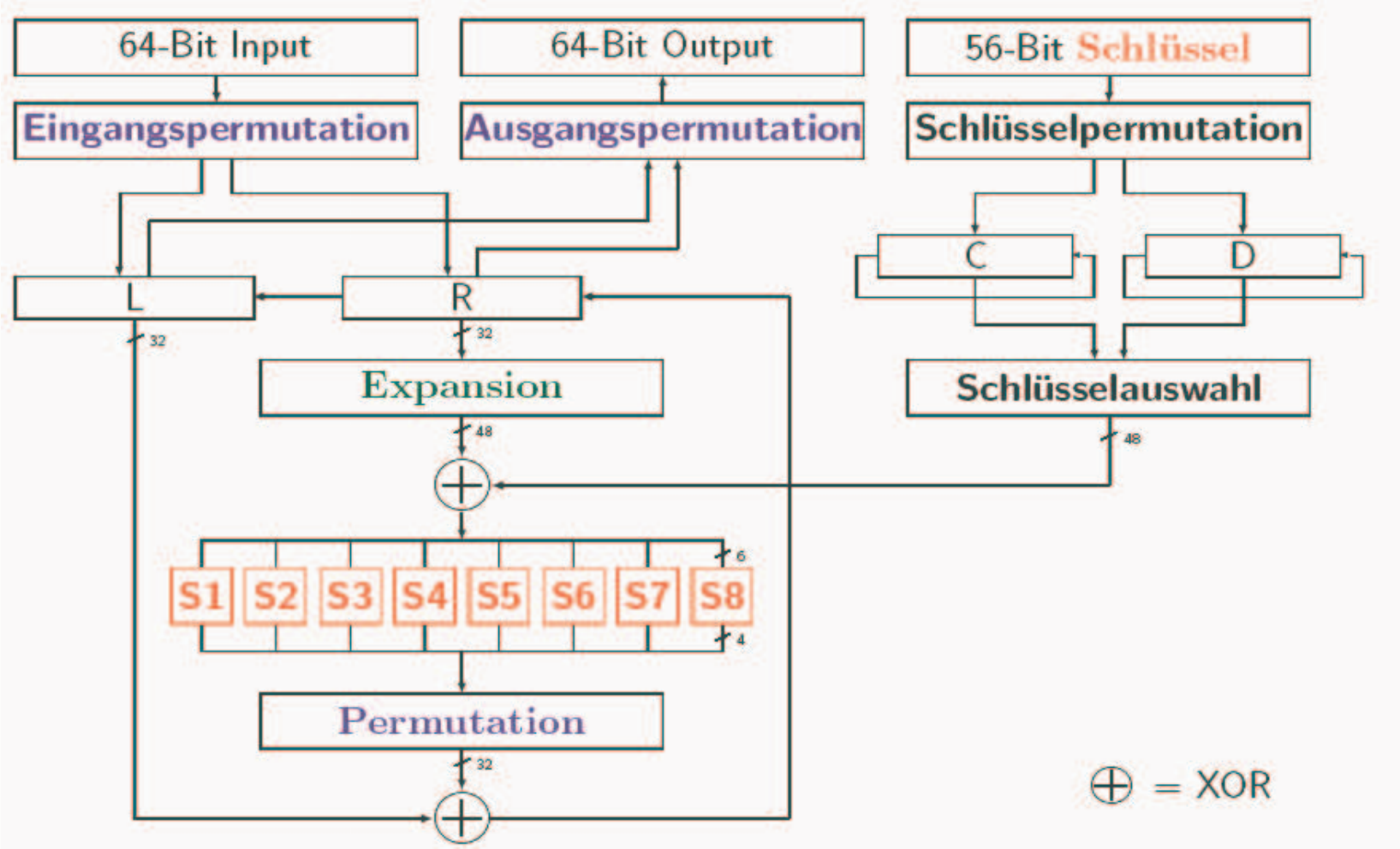
Data Encryption Standard

- Feistel-Chiffre:
  - Blockchiffre: Eingabeblock wird in zwei Hälften aufgeteilt.
  - Die Blöcke werden in mehreren Runden verarbeitet.
  - Die Rundenfunktion wird auf eine der beiden Hälften angewandt
  - und das Ergebnis mit der anderen Hälfte mittels XOR verknüpft.
  - Danach werden die beiden Hälften vertauscht und die nächste Runde wird ausgeführt.

Data Encryption Standard

- Diffusion und Konfusion durch Verknüpfung von Transpositionen und Substitutionen
- Blockchiffre mit 64 Bit Blocklänge
- Schlüssel von 64 Bit, 56 Bit frei wählbar
  → zu kurz!
- DES sowohl zum Verschlüsseln als auch zum Entschlüsseln: E = D
- Substitutionsboxen

Data Encryption Standard



Data Encryption Standard

Sicherheit des DES

- Gezielt entworfen, um differentielle Kryptoanalyse abzuwehren.
- Problem: kurze Schlüssel!
- Dreifachverschlüsselung mit zwei oder drei verschiedenen Schlüsseln: Triple DES (EDE)
- Effektive Schlüssellänge < 112 bit

Advanced Encryption Standard

- Wettbewerb ausgeschrieben in 1997.
- Am Anfang 15 Kandidaten.
- MARS (IBM), RC6 (RSA Labs), Rijndael (John Daemen u. Vincent Rijmen), Serpent (Anderson, Biham, Knudsen) und Twofish (Counterpane) wurden zur zweiten Runde zugelassen.
- Alle 5 ausreichend sicher.
- Rijndael gewann in 2000 aufgrund von Geschwindigkeit und einfachem Design.
- Seit 2001 Standard AES.

Eigenschaften

- Symmetrische Blockchiffre
- 128 Bit Blöcke und variable Schlüssellänge (128, 192 oder 256 Bit)
- Die Anzahl der Runden hängt von der Länge der Schlüssel ab (10,12 oder 14).
- Resistent gegen alle bekannten Methoden der Kryptoanalyse.
- Sehr schnell in Hard- und Software.

Funktionsweise

- Blocklänge z.B. 128 Bit
- State: 4x4 Matrix aus diesen 128 Bit

$A_{0,0}$	$A_{0,1}$	$A_{0,2}$	$A_{0,3}$
$A_{1,0}$	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
$A_{2,0}$	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$
$A_{3,0}$	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$

- Jedes Byte wird aufgefasst als Element in $GF(2^8)$ (Endlicher Körper mit 2^8 Elementen).

Funktionsweise

- SubBytes (S-Box): Byte-Substitution, genauer: Bilden des multiplikativen Inversen über GF(2^8)
- Shift Rows: zeilenweise Permutation
- Mix Columns: Multiplikation der Spalte mit einem festgelegten Polynom
- AddRoundKey: XOR-Verknüpfung mit Rundenschlüssel

Eigenschaften

- S-Box Substitution sorgt für Konfusion.
- Diffusion durch ShiftRow und MixColumn.
- Rundenschlüsselerzeugung aus geheimem Schlüssel ist nicht-linear und nicht invertierbar.
- Keine schwachen Schlüssel bekannt.
- Das Brechen von 1 bis 2 Runden ist einfach.
- Wie 5 Runden zu brechen sind, ist unbekannt.
- 10 Runden werden als nicht effizient (d.h. z.B. in weniger als 1 Jahr oder in weniger als 2^{128} Operationen) brechbar angesehen.

Kryptoanalyse

Klassifikation von Angriffen (bei bekanntem Verschlüsselungsverfahren):

- Ciphertext-only attack
  - Häufigkeit des Auftretens von bestimmten Buchstaben
- Known-plaintext attack
  - Standardbriefanfänge; Präambeln bei Programmen bzw. Kommunikationsprotokollen ...
- Chosen-plaintext attack
  - Z.B. Passwort-Cracking
- Chosen-ciphertext attack
  - Z.B. Angriff auf asymmetrische Verfahren

Methoden der Kryptoanalyse

- Brute force, Exhaustive Search
  - Durchprobieren aller Schlüssel
- Differentielle Kryptoanalyse (Biham, Shamir 1991)
  - Ermittlung von Unterschieden in den Chiffretexten abhängig von gezielt festgelegten Unterschieden in den Klartexten
- Lineare Kryptoanalyse (Matsui 1993)
  - basiert auf linearen Zusammenhängen zwischen Klartext, Chiffretext und Schlüssel

Hash-Funktionen

- Hash-Funktion
  - Recheneffiziente Funktion, die beliebig lange Binärstrings auf Binärstrings einer festen Länge, sogenannte Hash-Werte, abbildet.
- Kryptographische oder Einweg-Hash-Funktionen (one-way hash function)
  - Es ist rechnerisch unmöglich, zwei verschiedene Eingaben mit gleichem Hash-Wert zu finden (kollisionsfrei).
  - Es ist rechnerisch unmöglich, eine Eingabe zu einem gegebenen Hash-Wert zu finden.

Hash-Funktionen

- Kryptographische Verwendung bei
  - digitalen Signaturen (Hash-Wert der Nachricht wird signiert),
  - Datenintegrität (Virenschutz, Software-Verteilung),
  - Protokollen (z.B. Authentifizierung, digitale Signatur).
- A.k.a.: Fingerprint, kryptographische Prüfsumme, Message Integrity Check (MIC), Modification Detection Code (MDC), ...
  - Öffentlich bekannt
  - Kein geheimer Schlüssel
- Message Authentication Codes (MACs)
  - Geheimer Schlüssel
  - Datenintegrität und -authentizität
- Beipiele: MD2, MD5, RIPE-MD (128bit), SHA (160bit)

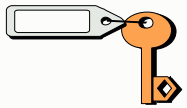
Kryptographische Systeme

- Symmetrische Verfahren (secret-key)
  - Identischer geheimer Schlüssel zur Ver- und Entschlüsselung
- Asymmetrische Verfahren (public-key)
  - Theoretisch beschrieben von Diffie und Hellman 1976.
  - Jeder Teilnehmer besitzt ein Schlüsselpaar (privater und öffentlicher Schlüssel).
  - Basis: Einweg-Funktionen (Funktionswertberechnung ist einfach, Umkehrung nur mit sehr großem Aufwand)

Public Key-Kryptographie

Teilnehmer (A und B)...

...und je Teilnehmer einen privaten....



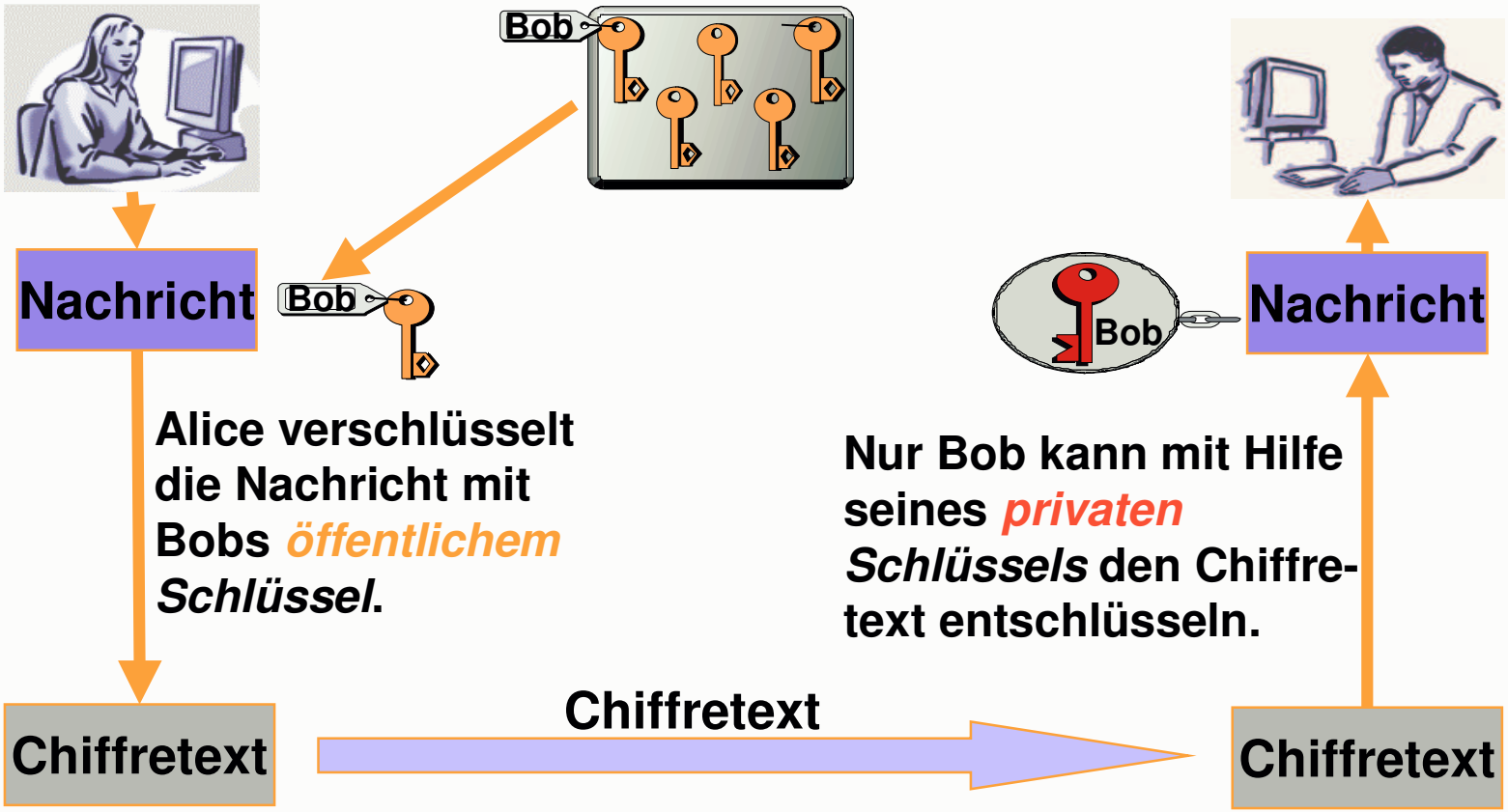
... und einen öffentlichen Schlüssel

Allgemeine Eigenschaften

- Die Schlüsselpaare (k_E, k_D) müssen folgende Eigenschaft erfüllen: Für alle Klartexte m muss gelten $D(E(m, k_E),k_D)) = m$, k_E öffentlich, k_D geheim.
- E und D sind einfach zu berechnen.
- k_D aus k_E nicht mit vertretbarem Aufwand berechenbar.
  - Einweg-Funktion mit Falltür
- Optional: $E(D(m,k_D),k_E)) = D(E(m,k_E),k_D)) = m$

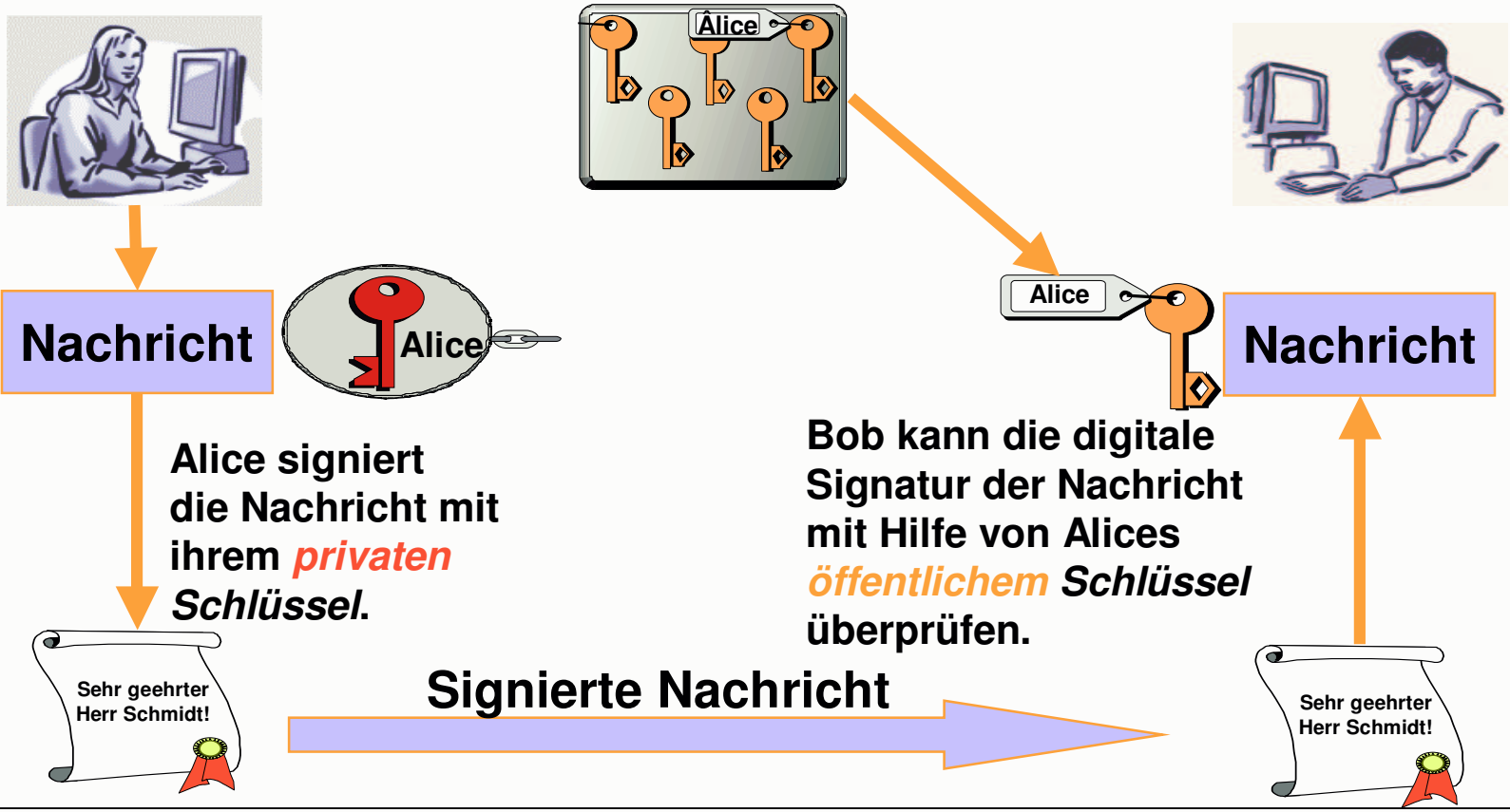
Verschlüsselung einer Nachricht

- Alice möchte Bob eine Nachricht senden.



Digitale Signatur

- Alice möchte eine Nachricht signieren und sicherstellen, dass die Nachricht an Bob nicht unbemerkt verändert werden kann.



Asymmetrische Kryptographie

- Verschlüsselung bewirkt Vertraulichkeit der Kommunikation.
- Digitale Signatur bewirkt Integrität und Authentizität der Nachricht.
- Mit Hilfe der Public Key-Kryptographie lässt sich auch eine Authentifizierung beim Logon erreichen.

Asymmetrische Systeme

- RSA (Signatur und Verschlüsselung)
  - Faktorisierung großer Zahlen
- Diffie-Hellman Schlüsselaustauschprotokoll
  - Diskrete Logarithmen ($y=a^x \bmod n$, y,a,n bekannt, was ist x?)
- DSA Signaturalgorithmus
  - Diskrete Logarithmen
- El Gamal Verschlüsselungsalgorithmus
  - Diskrete Logarithmen
- Algorithmen auf Elliptischen Kurven
  - Diskrete Logarithmen über elliptischen Kurven

RSA-Verfahren

- 1978 von Ronald Rivest, Adi Shamir und Leonard Adleman entwickelt.
- Mathematische Basis: Primfaktorzerlegung
- Einsatz: Verschlüsseln, Signieren, Schlüsselaustausch
- Standard-Verfahren (Andere Systeme nicht weit verbreitet.)
- Falls der Algorithmus mathematisch gebrochen werden sollte, sind viele Verschlüsselungssysteme gebrochen.

RSA-Verfahren

Mathematik:

- Restklassendivision: $x \bmod y = z \Leftrightarrow x = ky + z$, wobei x,y,z ganzzahlig
- Eulersche Zahl: $\varphi(m) = |\{a|ggt(a,m)=1 \wedge a<m\}|$
- Primzahl p: $\varphi(p) = p-1$
- Kleiner Satz von Fermat:
  - Falls $ggt(M,n)=1$, dann $M^{\varphi(n)} = 1 \bmod n$.

RSA-Verfahren

Vorbereitung:

- Generiere zwei große (und verschiedene) Primzahlen p und q und berechne das Modul n = pq.
- Es gilt: $\varphi(n) = (p-1)(q-1)$.
- Wähle d, $1 < d < \varphi(n)$, so dass $ggt(\varphi(n),d) = 1$.
- Berechne e, $1 < e < \varphi(n)$, mit $ed = 1 \bmod \varphi(n)$, d.h. e ist multiplikatives Inverses modulo $\varphi(n)$ zu d.
- (e,n) ist der öffentliche Schlüssel, (d,n) der private Schlüssel.
- d, p, q, $\varphi(n)$ sind geheim zu halten.

RSA-Verfahren

Verschlüsseln und Entschlüsseln:

- Transformation der Nachricht M in binäre Darstellung $M = M_1,...,M_m$, so dass für Blockgröße r gilt: $k = 2^r$ mit $k \leq n$.
- Blockweises Verschlüsseln mit Verschlüsselungsfunktion E: $E(M_i)=(M_i)^e \bmod n = C_i$
- Entschlüsselungsfunktion D: $D(C_i) = (C_i)^d \bmod n = M_i$

Beweis für Entschlüsselung

- Da $ed = 1 \bmod \varphi(n)$, gibt es k ganz mit $ed = 1 + k\varphi(n)$.
- Falls $ggT(M,p) = 1$, gilt nach kleinem Satz von Fermat $M^{p-1} = 1 \bmod p$.
- Somit $M^{1+k(p-1)(q-1)} = M \bmod p$.
- Dies gilt auch, falls $ggT(M,p) = p$.
- D.h. es gilt in jedem Fall $M^{ed} = M \bmod p$.
- Genauso: $M^{ed} = M \bmod q$.
- Da p und q verschiedene Primzahlen sind, folgt $M^{ed} = M \bmod n$.
- D.h. $D(C_i) = (C_i)^d \bmod n = (M_i)^{ed} \bmod n = M_i$

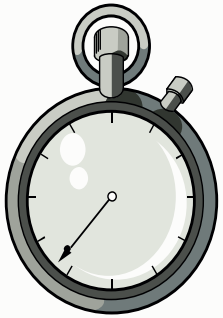
Beispiel

- $p = 47$, $q = 59$, $n = pq = 2773$, $\varphi(n) = 2668$
- $d = 157$, $e = 17$, dann ist $ed = 1 \bmod 2668$
- Verschlüsseln von $M = 920$:
  $C = E(M) = E(920) = 920^{17} \bmod 2773 = 948$
- Entschlüsseln von $C = 948$:
  $D(C) = D(948) = 948^{157} \bmod 2773 = 920$
- Bem.: Die Berechnung der Inversen e erfolgt mit dem erweiterten Euklidischen Algorithmus. Für die Berechnung der modularen Exponentialfunktionen gibt es ebenfalls effektive Algorithmen.

Sicherheit asymm. Verfahren

- Annahme: Ziehen von e-ten Wurzeln ist so schwer wie Faktorisierung des Moduls n (kein Beweis).
- Aufwand für Faktorisierung des Moduls nimmt stark mit der Größe des Moduls zu (kein Beweis, nur Erfahrung).
- Empfehlung zur Zeit: n mindestens 1024-stellige Binärzahl, p,q mind. 512-stellig.
- Für langfristige Verträge etc. n ca. 2048-stellig.
- Schlüssellänge bei El Gamal, DSA ähnlich.
- Elliptische Kurven bieten deutlich kürzere Schlüssellängen.

Asymmetrische Kryptosysteme



- **Problem: Relativ rechenzeitintensiv**
  z.B. RSA, elliptische Kurven

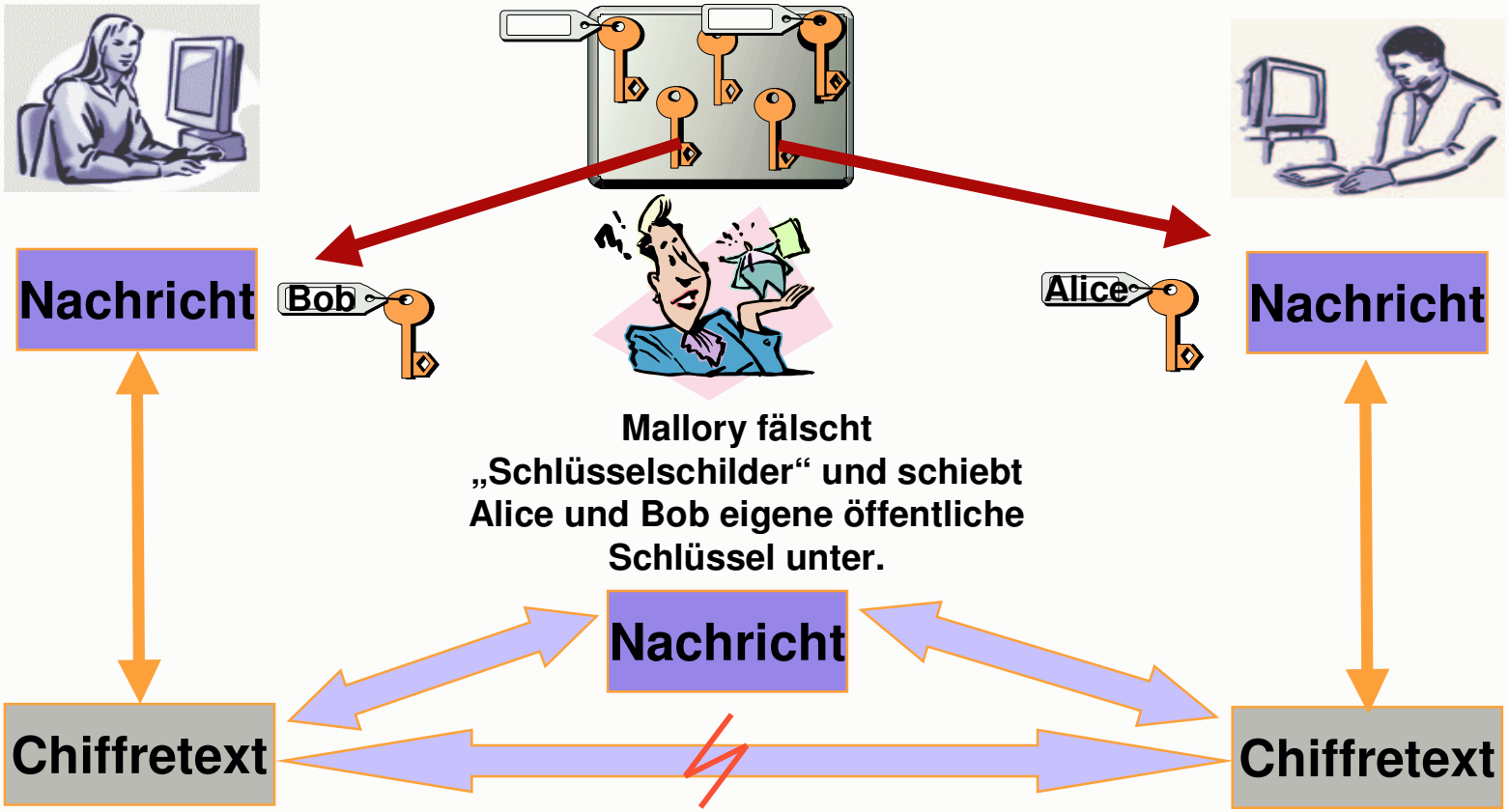
- **Keine geheime Schlüsselverteilung nötig!**
- **n Kommunikationspartner benötigen nur n Schlüsselpaare!**
- **Problem: Wer garantiert für Authentizität des öffentlichen Schlüssels?**

- **Digitale Signatur garantiert Authentizität und Integrität z.B. für Vertragsabschluss.**

Hybridverfahren

- Verbindung der Vorteile symmetrischer Verfahren (Schnelligkeit) mit den Vorteilen asymmetrischer Verfahren (Schlüsselverteilung)
- Konkret:
  - Alice und Bob tauschen zunächst ihre asymmetrischen Schlüssel aus.
  - Die asymmetrischen Schlüssel verschlüsseln und signieren nur einen zufällig bei Alice (oder Bob) erzeugten symmetrischen Sitzungsschlüssel.
  - Die tatsächliche Kommunikation wird dann mit dem Sitzungsschlüssel verschlüsselt.
- Alle heute auf asymmetrischen Protokollen aufsetzenden Anwendungen verschlüsseln in dieser Weise.

Man-in-the-Middle Attack

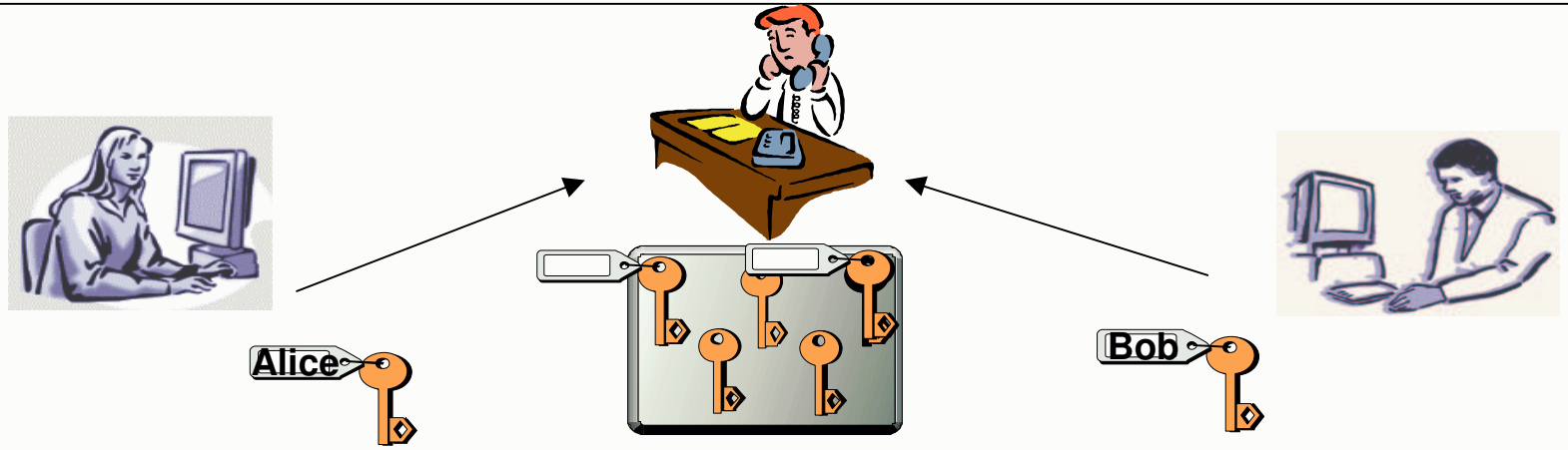


Folge: Mallory liest alle Nachrichten mit, ohne dass Alice und Bob dies bemerken!

Zertifikate

- Problem
  - Wer gibt die Sicherheit, dass die zum öffentlichen Schlüssel gehörenden Angaben zur Person vertrauenswürdig sind?
- Lösung
  - Eine Zertifizierungsstelle (Certification Authority, CA) verbürgt in einem Zertifikat mit der eigenen digitalen Signatur die Zusammengehörigkeit von personen-bezogenen Daten und öffentlichem Schlüssel.
  - Entscheidend für das Vertrauen in eine PKI sind die Certification Policies und deren Umsetzung in der Praxis.

Zertifizierungsinstanz (CA)



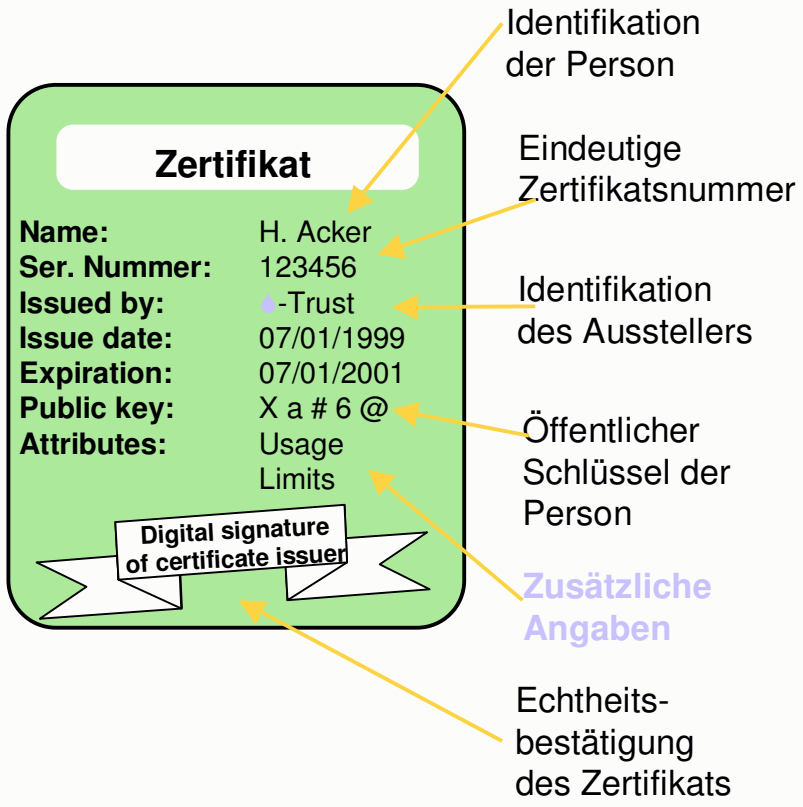
- Alice und Bob wenden sich mit ihren öffentlichen Schlüsseln an eine vertrauenswürdige dritte Instanz (CA), die die Schlüsselschilder mit ihrer eigenen Unterschrift signiert, d.h ein „Zertifikat“ erzeugt.
- Beide können dann anhand der Unterschrift der CA überprüfen, ob die Schlüssel authentisch ihrem Kommunikationspartner gehören.

Was enthält ein Zertifikat?

- Angaben zur Person (Inhaber)
- Öffentlicher Schlüssel
- Seriennummer des Zertifikats
- Gültigkeitsdauer
- Angaben zum Aussteller (CA)
- Digitale Signatur der CA
- Erweiterungen

Zertifikate

- **Zertifikate**
  - verknüpfen natürliche/juristische Personen mit kryptographischen Schlüsseln,
  - sind zeitlich begrenzt,
  - können zusätzlich persönliche Angaben der Zertifikatsinhaber enthalten.
- Zertifikate werden von **vertrauens-würdigen Stellen** ausgegeben: (Trusted Third Party, Certification Authorities)
  - Spezifische Zertifikatsangaben und proprietäre Erweiterungen
  - Unterschiedliche Praxis der Vergabe von Zertifikaten (Certification Policies)
  - Kooperationen zwischen eigenständigen Certification Authorities
- Nutzung digitaler Signaturen setzt zwingend *Zertifikate* voraus.



Public Key-Infrastruktur (PKI)

- **Die für eine Kommunikation mit asymmetrischen Schlüsseln nötige Infrastruktur wird "PKI" genannt, z.B. zertifikatsbasiert:**
- Certification Authority (CA) stellt Zertifikate aus.
- Registration Authority (RA) verwaltet Zertifikate.
- End Entity: der Zertifikatsinhaber (Person, Server, Organisation).
- Verzeichnisdienst (Directory Service) veröffentlicht Zertifikate.
- Kartenproduzent stellt sichere Schlüsselspeicher bereit (Smartcards, USB-Token, etc.).

Public Key-Infrastruktur (PKI)

- Anforderungen an eine CA:
  - Interoperabilität (Standards)
  - Dezentrale Verwaltung (online)
  - Revozierungssystem
  - Verlängerung von Zertifikaten
  - Anbindung an Anwendungssysteme
  - Support für unterschiedliche Zertifikatstypen
  - Kommunikation mit Partner-CAs
- Alternative zur CA: Web of Trust
  - Bekannte Kommunikationspartner vertrauen sich gegenseitig und unterschreiben Zertifikate ohne eine zentrale Stelle.
  - Realisiert im sicheren Email-Standard PGP (Pretty Good Privacy).

Public Key-Infrastruktur (PKI)

- Zertifikatstypen: X.509, PGP, WTLS
- Überprüfung von Revozierungen
- Sperrlisten (Certificate Revocation List, CRL)
- Online-Abfrage (Online Certificate Status Protocol, OCSP)
- Schutz des privaten Schlüssels:
  - Passwort-geschützte Datei, SmartCard, USB-Token, ...

Anwendungen

- E-Mail
- E-Commerce
- Elektronische Bankgeschäfte (HBCI, Elko)
- Verbindliche Verträge (Signaturgesetz)
- Online-Behördengänge (Signaturgesetz)
- Signatur von Software (z.B. Java-Applets)
- Virtual Private Networks (VPN)
- Mobile Dienste

Diffie Hellman

- Beruht auf diskretem Logarithmusproblem über M = GF(q), q große Primzahl, d.h. $a = b^x \mod q$ ist für große q schwer lösbar.
- Wähle primitive Einheitswurzel a in M, d.h. jedes Element aus M lässt sich als Potenz von a darstellen.
- a findet man durch zufällige Wahl mit nachfolgender Überprüfung (einfach, falls $q-1 = 2^r$).
- q und a werden veröffentlicht!

Diffie Hellman

- Teilnehmer A und B wählen Zufallszahl X_A, X_B (zwischen 1 und q-1).
- X_A ist der geheime Schlüssel von A, X_B analog.
- $Y_A = a^{X_A} \bmod q$ ist der öffentliche Schlüssel von A, B analog.
- Zur Kommunikation berechnet jeder Teilnehmer den Sitzungsschlüssel

$$K_{AB} = K_{BA} = a^{X_A X_B} \bmod q = Y_A^{X_B} \bmod q = Y_B^{X_A} \bmod q$$

Diffie Hellman

- Protokoll zur Vereinbarung eines Sitzungsschlüssels ohne gemeinsame Geheimnisse.
- Keine Authentifizierung der Partner.
- Man-in-the-Middle-Angriff möglich.

Schlüsselverwaltung

- Die Sicherheit jedes kryptographischen Systems beruht auf der sicheren Verwaltung der geheimen Schlüssel.
  - Was nützt z.B. ein ausgefeilter kryptographischer Algorithmus, wenn die Benutzer PINs auf Chipkarten schreiben?
- Probleme:
  - Schlüsselerzeugung
  - Schlüsselspeicherung
  - Schlüsselvernichtung

Schlüsselerzeugung

- Symmetrisch:
  - Gute Zufallszahlengeneratoren notwendig.
  - Meistens realisiert: Pseudozufallszahlengeneratoren.
  - Weitere Methoden: Würfeln, Münze werfen, atmosphärisches Rauschen, radioaktiver Zerfall, Mausbewegung, Tastaturanschlagszeiten.
- Asymmetrisch:
  - RSA: große Primzahlen notwendig.
  - Tatsächlich eingesetzt: Pseudoprimzahlen, d.h. man stellt mit einer gewissen Wahrscheinlichkeit fest, ob es sich um eine Primzahl handelt.

Schlüsselspeicherung

- Gedächtnis
  - Wenig zuverlässig.
- Dateisystem
  - Kopierbar.
  - Kann über PIN geschützt werden.
- Hardware
  - Chipkarte, USB-Token, PCI-Karte.
  - Kann über PIN geschützt werden.
  - Nicht einfach kopierbar.
- Verteilung auf mehrere Schlüsselinhaber

Schlüsselvernichtung

- Oftmals vernachlässigt.
- Auch aktuell nicht mehr verwendete Schlüssel können für einen Angreifer nützlich sein, falls er in den Besitz von altem Chiffretext gekommen ist!
- Geheimhaltungsfristen sind zu beachten.

Recovery

- Warum?
  - Ausscheiden von Mitarbeitern, mit deren Schlüssel verschlüsselt wurde.
  - Vertretungsregelungen
  - Schlüsselverlust
  - Geheimdienstinteresse
- Wie?
  - Schlüsselrückgewinnung (Key Recovery)
  - Schlüsselhinterlegung (Key Escrow)
  - Nachrichtenrückgewinnung (Message Recovery)

Netzwerksicherheit

- Kapitel 4: Netzwerksicherheit
  - Überblick über TCP/IP
  - Schwachstellen und Bedrohungen IP-basierter Protokolle und Dienste
  - Sichere elektronische Kommunikation
  - Firewalls

Netzwerksicherheit

Sicherheitsziele:

- Integrität und Authentizität der übertragenen Daten
- Vertraulichkeit der übertragenen Daten
- Verbindlichkeit
- Verfügbarkeit

Netzwerksicherheit

Schwachstellen, Bedrohungen

- Netzwerkprotokolle meist nicht für offene Netze mit Blick auf Sicherheit entwickelt.
  → auch Internet-Protokollsuite IPv4
- Bieten vielfältige Möglichkeiten für
  - Abhören (Sniffing, Snooping) und
  - Manipulation der übertragenen Daten,
  - Vortäuschen falscher Identität (IP-, DNS-, Web-Spoofing, Email),
  - Man-in-the-Middle-, Replay-Attacken,
  - Denial-of-Service-Attacken,
  - anwendungsspezifische Angriffe, z.B. auf Web-Anwendungen (SQL-Injection, Cookie-poisoning, Buffer-overflow, Manipulation von Parametern, ...).

Netzwerksicherheit

Sicherheitsmaßnahmen:

- Physikalische Zugangskontrolle
- Netzwerkzugangskontrolle
- Netztrennung, Firewall-Gateways (Paketfilter, Proxy)
- Virenschutz
- Kryptographische Schutzfunktionen
  - S/MIME, PGP, PEM, ...
  - TLS/SSL
  - IPsec
- Authentifizierungssyteme
  - Passwort-basiert, kryptografisch, Tokens, Biometrie
  - Kerberos

Überblick über TCP/IP

- Suite von Kommunikationsprotokollen
- Namesgeber sind dabei
  - TCP = Transmission Control Protocol
  - IP = Internet Protocol
- Entwickelt von U.S. Defense Advanced Research Projects Agency
- ARPANET seit 1983
- Information über TCP/IP-Protokolle ist als Requests for Comments (RFC) veröffentlicht.

Eigenschaften von TCP/IP

- Offene Protokollstandards, frei erhältlich, hardware- und betriebssystem-unabhängig.
- Unabhängig von spezifischer physikalischer Netzwerk-Hardware (Ethernet, Token Ring, Wählleitung, X.25-Netz, ...).
- Gemeinsames Adressschema.
- Standardisierte High-level-Protokolle für weit verbreitete Anwenderdienste.

OSI Referenzmodell

- Architekturmodell für Beschreibung von Struktur und Funktion von Datenkommunikationsprotokollen
- ISO/OSI (International Standards Organization/Open Systems Interconnect)
- Sieben Schichten (layers) mit definierten Datenkommunikationsfunktionen
- Stack oder Protocol Stack

OSI Referenzmodell

7 Application Layer	Besteht aus Anwendungsprogrammen, die das Netzwerk benutzen.
6 Presentation Layer	Standardisiert Datendarstellung für die Anwendungen.
5 Session Layer	Verwaltet Sitzungen zwischen Anwendungen.
4 Transport Layer	Liefert Ende-zu-Ende-Fehler-erkennung und -behebung.
3 Network Layer	Verwaltet Verbindungen über das Netzwerk für höherliegende Schichten.
2 Data Link Layer	Bietet verlässliche Datenauslieferung über die physikalische Verbindung.
1 Physical Layer	Definiert die physikalischen Eigenschaften des Netzwerkmediums.

OSI Referenzmodell

- Daten werden den Stack hinunter durchgereicht, bis sie von den Physical Layer Protokollen über das Netzwerk übertragen werden.
- Kommunikationspartner (peer) muss gleichen Stack implementiert haben.
- Daten werden am anderen Ende wieder den Stack hochgereicht zur empfangenden Anwendung.
- Schichtenmodell minimiert Auswirkungen von technologischen Änderungen auf die gesamte Protokollsuite.

TCP/IP Protokollarchitektur

4 Application Layer (SMTP, Telnet, FTP etc.)	Besteht aus Anwendungen und Prozessen, die das Netzwerk benutzen.
3 Host-to-Host Transport Layer (TCP, UDP, ICMP)	Leistet Ende-zu-Ende-Datenauslieferung.
2 Internet Layer (IP)	Definiert die Datagramme und handhabt das Routing der Daten.
1 Network Access Layer (Ethernet, FDDI, ATM etc.)	Besteht aus Routinen für den physikalischen Netzwerkzugang.

TCP/IP Protokollarchitektur

- **Encapsulation**
  - Jede Schicht im Stack fügt den zu sendenden Daten Kontrollinformationen hinzu, sogenannte **Header**.
  - Umgekehrt werden beim Empfang Header entfernt.
- **Network Access Layer**
  - Encapsulation of IP datagrams into the frames transmitted by the network.
  - Mapping of IP addresses to the physical addresses used by the network.
  - Address Resolution Protocol (ARP)

IP

- **Internet Layer**

  Internet Protocol (RFC 791)

  – Datagramm: Grundeinheit für Übertragung im Internet

  – Internet Adressschema (32-bit IP-Adresse)

  – Routing von Datagrammen zu entfernten Hosts über Gateways

  – Fragmentierung und Reassemblierung von Datagrammen

  – Verbindungsloses Protokoll

  – Nicht-verlässliches Protokoll (keine Fehlererkennung und -behebung)

ICMP

- **Internet Layer**

  Internet Control Message Protocol (ICMP), RFC 792

  – Flusskontrolle
  – Erkennung unerreichbarer Ziele
  – Umleitung von Routen (ICMP Redirect Message)
  – ICMP Echo Message (UNIX ping)

UDP

- **Transport Layer**

  User Datagram Protocol (UDP)

  - Verbindungslose Datagramm-Auslieferung
  - Nicht-verlässlich
  - Minimaler Protokoll-Overhead
  - 16-bit Quell- und Zielport
    - Identifizieren zuständige Anwendung auf beiden Hosts

TCP

- **Transport Layer**

  Transmission Control Protocol (TCP)

  – Verlässliche Datenauslieferung
    - Segmente mit Sequenznummern
    - Positive Acknowledgment with Retransmission
    - Ende-zu-Ende Fehlererkennung und -behebung
  – Verbindungsorientiert
    - Three-way handshake
    - SYN, ACK und FIN flags
    - ACK-Segment liefert auch Flusskontrolle
  – Byte-stream
  – 16-bit Quell- und Zielport

Anwendungsprotokolle

- **Application Layer**

Weit verbreitete Anwendungsprotokolle:

- **Telnet** (Network Terminal Protocol)
- **FTP** (File Transfer Protocol)
- **SMTP** (Simple Mail Transfer Protocol)
- **DNS** (Domain Name System)
  - Ordnet IP-Adressen Namen zu.
- **NFS** (Network File System)
- **HTTP** (Hypertext Transfer Protocol)
  - World Wide Web
- Routing Protokolle (RIP, OSPF, BGP-4, ...)

TCP/IP Protokollarchitektur

- Adressierung
  - IP-Adressen identifizieren Hosts im Internet.
  - Bsp.: 192.178.16.1
  - Netzwerk- und Host-Adressteil
    - Class A, B, C, D, E
    - Default route, loopback address
    - Broadcast address
    - Multicast
    - Subnets
- Routing
  - Gateways liefern Daten an das korrekte Netzwerk ab.
- Multiplexing
  - Protokoll- und Port-Nummern liefern Daten an das korrekte Software-Modul im Host ab.

Schwachstellen und Bedrohungen IP-basierter Protokolle und Dienste

Beispiele ohne Anspruch auf Vollständigkeit:

- Lauschangriffe
- ARP-Spoofing
- IP-Spoofing, UDP-Spoofing
- DNS-Schwachstellen
- TCP
  - Sequence number guessing
  - Session hijacking
- Source-Routing-Angriff
- Tiny-Fragment-Angriff

Schwachstellen und Bedrohungen IP-basierter Protokolle und Dienste

- ICMP-Angriffe
- Denial-of-Service-Angriffe (DoS)
- Distributed DoS
- r-Kommandos
- ...

Lauschangriffe

- Engl.: **sniffing** (Schnüffeln)
- **Mitlesen** der übertragenen Daten.
- Leicht möglich in shared-media LANs (z.B. Ethernet, Token Ring).
- Paket-Sniffer, Netzwerkmonitore (snoop, tcpdump, ethereal, ...)
- Netzwerkkarte in promiscous mode betreiben.
- Switches bieten keinen echten Schutz (s. ARP-Spoofing).

Lauschangriffe

- Hohe Bedrohung, da viele Anwendungs-protokolle Nutzdaten im **Klartext** übertragen.
- Oft sind **UserID** und **Passwort** das Ziel.
- Beispiele: FTP, Telnet, POP3, HTTP

ARP-Spoofing

- Adresstabelle im ARP-Cache von Routers und Hosts `(arp -a)`
- ARP-Request: Broadcast an alle Rechner im Netzsegment
- ARP-Reply von Rechner mit angefragter IP-Adresse
- Zustandsloses Protokoll: Es werden auch Antworten ohne (eigene) Anfragen akzeptiert.

ARP-Spoofing

- Engl.: Spoofing (Schwindeln, Vortäuschen)
- ARP-Spoofing: Angreifer schickt gefälschten ARP-Reply an Opfer.
- Ermöglicht auch das Mitschnüffeln in geswitchten **LAN**s, indem der Angreifer den gesamten Netzverkehr über seinen Rechner umleitet (Man-in-the-Middle).
  - Sniffer: dsniff, ettercap, ...

IP-Spoofing

- Versenden von Datagrammen mit gefälschten IP-Quell-Adressen.
- Ziel: Erlangen von Privilegien eines vertrauenswürdigen Systems.
- Besonders einfach beim verbindungslosen UDP, abhängig von Anwendungsprotokoll.
  - Beispiel: DNS-Anfrage besitzt eine Query-Id (16 bit). Es werden im Schnitt 32768 Antworten benötigt, um die Query-Id zu raten.
- TCP besitzt ACK und Sequenznummer.
  - Wird später behandelt.

DNS

- Domain Name System (1983 erfunden)
- Nachschlagedienst: verteilt, global skalierend, redundant, kohärent.
- Übersetzt Namen in Adressen
  - bzw. irgendwas in irgendwas anderes.
- Beispiele:
  - Domänenname www.uni-hildesheim.de in IP-Adresse 147.172.16.46 (Resource Record „A“)
  - Mail Exchanger für Domäne uni-hildesheim.de ist rzm31.rz.uni-hildesheim.de (RR „MX“)

DNS

- Baumstruktur mit Delegation der administrativen Kontrolle an Unterbäume.
- Root des DNS wird als „.“ referenziert.
- Root Service wird durch 13 IP-Adressen bereitgestellt.
- Top Level Domains:
  - .com, .gov, .edu, .arpa, .de, .uk ...

DNS

- **Resolver:** Client-seitiger Code, der DNS-Anfragen an Caching-Server stellt und auf Antworten wartet.
- **Caching Server:** Proxy, der Daten für Resolver holt und zwischenspeichert.
- **Authoritative Server:** veröffentlicht Zonendaten.
  - **Primary**
  - **Secondary**
- Port 53 (**UDP** und TCP)

DNS-Sicherheit

- DNS-Spoofing bietet Angreifern die Möglichkeit Internet-Nutzer auf gefälschte Sites zu lenken.
- **Reichweite** des DNS-Angriffs ist dabei potentiell das ganze Internet.
  - Vgl. ARP-Spoofing: lokales Netzwerk.
- Interessantes Ziel sind die Domänennamen von e-Commerce-Anbietern auf DNS-Servern von Internet Service Providern.
- **DNS ist ein kritischer Teil der Internet-Infrastruktur.**

DNS-Protokollschwachstellen

Angriffsszenarien:

- Sich als ein Primary DNS-Server ausgeben (Impersonation).
  - Den Secondary Server dazu bringen, den eigenen DNS-Server statt des wahren Primary Server zu verwenden, z.B. durch Cache Poisoning.
- Sich als ein Caching Server ausgeben (Cache Impersonation).
  - Z.B. durch DHCP-Angriff
  - oder Fälschen von Antworten (im Durchschnitt 32768 Antworten, um die Query-Id zu raten).

DNS-Protokollschwachstellen

Angriffsszenarien (Fortsetzung):

- Korrumpieren des Caches (Cache Pollution)
  - Einige ältere DNS-Software (z.B. BINDv4, Microsoft DNS) prüft nicht die Query-Id.
  - Anfrage abfangen und schneller antworten.
  - Opfer mit Antworten fluten.
- Unautorisierte Updates bei Verwendung von dynamischem DNS.
  - Ggf. Source-IP-Adresse fälschen.

DNS-Infrastruktursicherheit

- Root Name Server können „single point of failure“ darstellen.
  - Oktober 2002: Ping-flood-Angriff, 9 von 13 Root Name Servern waren „unten“
- BIND (Berkeley Internet Name Domain server) Monokultur.
- Modifikationen des DNS-Cache aus politischen Gründen.
- Hacking der Registrierungsstellen/Registrare.
- Buffer-Overflow-Schwachstelle in BIND Resolver-Bibliothek.

TCP-Sicherheit

Sequence Number Guessing

- Sicherheit der TCP-Verbindung beruht
  - auf Three-way Handshake beim Verbindungsaufbau
  - und damit auf korrekter Initial Sequence Number (ISN).
- Vorhersage der ISN ermöglicht Angriff.
- Ausnutzen von TCP-Implementierungsschwächen:
  - ISN wird durch einfachen Algorithmus generiert.
    - RFC 793 spezifiziert, dass die ISN als 32-bit-Zähler angesehen werden soll, der alle 4 Mikrosekunden um 1 erhöht wird.
  - Pseudo-Zufallszahlengenerator
    - Ermöglichen teilweise das Erraten der ISN nach einigen Beobachtungen legitimer Verbindungsaufbauten.
- Verhindern, dass vorgetäuschter Host mit Reset-Paket antwortet.

TCP-Sicherheit

Session Hijacking

- Einseitiges Einschleusen eines Kommandos, z.B. in eine Telnet-Sitzung.
- Oder komplette Übernahme einer bestehenden TCP-Verbindung.
- Client wird durch Einfügen von Paketen desynchronisiert.
- Symptom: ACK-Sturm
- *juggernaut*, Phrack magazine, issue 50

Source-Routing-Angriff

- IP-Option: strict bzw. loose source routing
- Sender gibt die Route vor.
- Lässt sich für Angriffe gut ausnutzen.
- Security Gateways (Firewalls) müssen Pakete mit Source-Routing-Option zurückweisen.

Tiny-Fragment-Angriff

- IP-Header enthält sicherheitsrelevante Daten, insbesondere IP-Adresse und -Port von Quelle und Ziel.
- Bei Fragmentierung eines Datagramms mit sehr kleinen Fragmenten passen diese nicht mehr ins erste Fragment.
- Paketfilter werten ggf. nur erstes Fragment aus.

ICMP-Angriffe

- Großer Funktionsumfang von ICMP:
  - Flusskontrolle, Umleitung von Routen, ... (s.o.)
- Bietet vielfältige Angriffswerkzeuge.

Beispiele:

- Echo Request und Echo Reply
  - `ping` gibt Auskunft über interne Netzwerkstruktur.
  - Ping-Flooding
- Destination Unreachable
  - Ältere ICMP-Implementierungen analysieren nicht den Header des fehler-auslösenden Datagramms.
  - Eine (gefälschte) Nachricht kann alle Verbindungen zu einer IP-Adresse unterbrechen.

ICMP-Sicherheit

- Source Quench
  - Fehlernachricht kann generiert werden, wenn Datagramme mit zu hoher Rate empfangen werden.
  - Kann für Denial-of-Service missbraucht werden.
- Redirect
  - Wird von Router an Absender eines Datagramms geschickt, wenn dieser einen anderen Router nutzen sollte.
  - Datenverkehr lässt sich umlenken.
  - Lauschangriff, Man-in-the-Middle, Denial-of-Service, ...

ICMP-Sicherheit

- Ping-of-Death
  - Übergroßer ICMP Echo Request.
    - `ping -t -l 65510 <IP-address>`
  - Führt in einigen Implementierungen zum Systemabsturz.
- Tunneling
  - Paketfilter prüfen ggf. nicht die Nutzdaten von ICMP Nachrichten.
  - Gelingt es Angreifer einen ICMP-Server auf Rechner im internen Netz zu installieren, kann er den Paketfilter tunneln.

Denial-of-Service-Angriffe

- Angriffe gegen Verfügbarkeit eines Systems/einer Anwendung.
- Ausnutzung von Schwachstellen der verschiedenen Ebenen der Netzwerkarchitektur.
- Hohe Gefahr:
  - Oft leicht durchzuführen.
  - Schutz- und Gegenmaßnahmen nur schwer möglich.
- Meist geringes Schadenspotential. Aber:
  - E-Commerce-Anbieter?
    - Angriff auf Web-Portale wie Yahoo, Amazon, eBay im Februar 2000
  - Angriff auf Root-Name-Server Oktober 2002
  - Kritische Infrastruktur?

Denial-of-Service-Angriffe

Beispiele:

- Verschiedene ICMP-Angriffe (s.o.)
  - Z.B. Ping-Flooding
- Smurf
  - ICMP Echo Request mit IP-Broadcast-Adresse als Ziel und gefälschter Quell-IP-Adresse des Opfers
- ARP Broadcast Storms
  - Beispielszenario:
    - Angreifer sendet Datagramm an nicht-existente Adresse als Broadcast-Frame.
    - Alle Router antworten mit ARP-Request.
    - Angreifer sendet ARP-Reply mit Broadcast-Adresse.
    - Paket wird mit Broadcast an alle Rechner geschickt.

Denial-of-Service-Angriffe

- UDP-Flooding
- SYN-Flooding
  - Überfluten des Servers mit Anfragen zum Verbindungsaufbau (SYN-Segment).
  - Ggf. gefälschte Absender-Adresse/-Port (IP-Spoofing)
  - Belegung von Speicher in der Verbindungs-Queue
  - Freigabe erst nach Time-out

Distributed Denial-of-Service

- Angriff von großer Anzahl von verteilten (distributed) Systemen im Internet.
- Bekannt seit Mitte 1999.
- Angreifer nutzt gehackte Systeme im Internet und bereitet Angriff dort vor.
- Mehrere Stufen zwischen Angreifer und Opfer erschweren die Rückverfolgung.
- Frei im Internet verfügbare Tools, wie
  - Trin00,
  - Tribe Flood Network,
  - Stacheldraht.
- Wechselnde Protokolle für Angriffsdatenstrom und IP-Spoofing erschweren Gegenmaßnahmen.

r-Kommandos

- r-Kommandos unter Unix (`rlogin`, `rsh`) ermöglichen Zugriff auf entfernten Rechner ohne weitere Passworteingabe.
- Überprüft werden nur IP-Adresse und Login-Name auf dem Client.
- Dateizugriffsrechte auf Konfigurationsdateien `/etc/hosts.equiv` und `.rhosts` beachten!
- Werden dort Domänennamen statt IP-Adressen eingetragen, führt DNS-Spoofing zum Ziel.

Sichere elektronische Kommunikation

- Beispiele: Email, WWW und IP
- Exemplarisches Aufzeigen der oben genannten Schwachstellen und Bedrohungen für diese Protokolle/Anwendungen
- Kryptographische Schutzfunktionen
  - **Digitale Signatur** für Integrität und Authentizität der Kommunikation
  - **Verschlüsselung** für Vertraulichkeit der Kommunikation
  - Einsatz in den verschiedenen Schichten: Anwendung, Transport, Internet
- Firewalls und Authentifizierungssysteme: später

SMTP

- Beispielablauf

```
220 fwd05.sul.t-online.com T-Online ESMTP receiver fsmtpd ready.
HELO me.foo.bogus
250 Ok.
MAIL FROM:<george@whitehouse.gov>
250 Ok.
RCPT TO:<fricke@uni.bogus>
250 Ok.
DATA
354 Ok, start with data.
From: george@whitehouse.gov
To: fricke@uni.bogus
Subject: Mail spoofing

Wie geht das?
.
250 Message accepted.
```

Sichere Email?

- Schwachstellen, Bedrohungen
  - Absenderadresse und Nachricht leicht fälschbar
  - Nachricht abhörbar und manipulierbar
  - Store-and-Forward-Prinzip
  - Denial-of-Service-Angriffe (auf empfangende Mailsysteme)
  - Spam
  - Mail relaying
  - ...
- Sicherheitsanforderungen
  - Vertraulichkeit der Nachricht
  - Authentizität der Herkunft
  - Integrität der Nachricht
  - Non-repudiation
  - Einwurfbestätigung
  - Empfangsbestätigung
  - ...

Sichere Email

- **Kryptographische** Sicherheitsdienste für elektronische Nachrichten
  - Authentizität, Integrität und Verbindlichkeit durch **digitale Signaturen**
  - Ende-zu-Ende-Vertraulichkeit und Datensicherheit durch **Verschlüsselung**
- Standards/Produkte
  - S/MIME
  - PGP
  - PEM (veraltet)
  - Lotus Notes
  - ...

Sichere Email

- Probleme, Randbedingungen
  - Welches Produkt, welchen Standard nutzt der Kommunikationspartner? (Interoperabilität)
  - Anforderungen an Schlüssel-, Zertifikatsmanagement; PKI
  - Message Recovery
    - Firmeninteresse vs. Vertraulichkeit
  - Die Anforderungen, insbesondere an „Interoperabilität“ und „PKI“, hängen stark vom Anwendungsszenario ab:
    - Unternehmensinterne Kommunikation
    - Kommunikation mit Partnern (Lieferanten, Dienstleister,...), B2B
    - Kommunikation mit (bekannten/spontanen) Kunden (E-commerce), B2C
  - Verteilerlisten
  - Vertretungen (insbesondere in Firmen)
  - ...

S/MIME

- **Secure/Multipurpose Internet Mail Extensions**
  S/MIME Version 3 (Proposed Standard)
  - Message Specification (RFC 2633)
  - Certificate Handling (RFC 2632)
- Senden und Empfangen von sicheren MIME-Daten
  - I.d.R. durch MUAs (mail user agents), wie Netscape Messenger, MS Outlook u.a.
  - Aber nicht auf Mail beschränkt.
- Basiert auf „Cryptographic Message Syntax“ (RFC 2630, abgeleitet von PKCS #7) und MIME-Specs
  - Benutzt die Content Types Data, SignedData und EnvelopedData.
  - Benutzt X.509-Zertifikate (PKIX).

Enveloped-only Message

Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V

Signed-only Message (1)

Content-Type: application/pkcs7-mime; smime-type=signed-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75

Signed-only Message (2)


```
Content-Type: multipart/signed;
        protocol="application/pkcs7-signature";
        micalg=sha1; boundary=boundary42

--boundary42
Content-Type: text/plain

This is a clear-signed message.

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

--boundary42--
```

S/MIME

- Signatur und Verschlüsselung gleichzeitig möglich.
- Absender-Zertifikat kann signierter Nachricht beigepackt werden.
- ➢ Empfänger kann dann an Absender verschlüsselte Nachrichten senden.
- ➢ Gängige MUAs, wie Netscape Messenger, MS Outlook, haben dies benutzerfreundlich implementiert.
- ➢ Statt Peer-to-peer-Zertifikatsaustausch auch Einsatz eines Verzeichnisdiensts möglich.

HTTP

- Beispielablauf

```
GET / HTTP/1.1
Host: www.uni-hildesheim.de

HTTP/1.1 200 OK
Date: Wed, 18 Dec 2002 20:00:34 GMT
Server: Apache
Last-Modified: Wed, 18 Dec 2002 08:58:56 GMT
ETag: "edf9b-3425-3e0038d0"
Accept-Ranges: bytes
Content-Length: 13349
Content-Type: text/html


<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html><!-- #BeginTemplate "/Templates/generic.template.dwt" -->
...
```

HTTP

- Anwendungsprotokoll über TCP (RFC 2068)
- **Zustandsloses Klartextprotokoll**
- HTTP **Header** enthält alle wichtigen Informationen.
  - Was, Wie, Wer, von Wo
- Wichtigste Methoden: **PUT** und **GET**
- **GET:** Daten als URL-Argumente versendet.
- **PUT**: Daten als Inhalt versendet.
- Alle Variablen und Werte sind **manipulierbar** (Formularfelder, Cookies, URL).
- **Session-Id** beseitigt Zustandslosigkeit für Web-Anwendungen.
  - Cookies
  - URL rewriting

World Wide Web (WWW)

- Schwachstellen, Bedrohungen
  - **Abhören** der übertragenen Daten
    - Passwörter, Kreditkartennummern, persönliche Daten, Session-Id, ...
  - **Manipulation** der übertragenen Daten
  - **Proxies** (z.B. für Caching)
  - **Vortäuschen**/Fälschen einer Web-Site
  - Aktive Inhalte (JavaScript, ActiveX, Java, ...)
    - **Malicious Code**

World Wide Web (WWW)

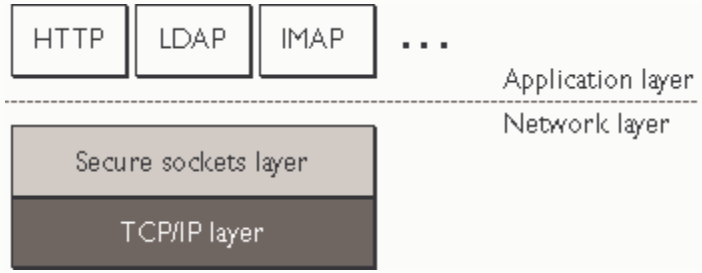
- Schwachstellen, Bedrohungen (cont.)
  - Spezifische **Angriffe auf Web-Anwendungen**:
    - Manipulation von Eingabe-Parametern (z.B. hidden fields, cookies, vorbereitete Links)
    - Directory Traversal
    - Encoding-Attacken (Unicode Exploit)
    - Cross-site-scripting (XSS)
    - Buffer-overflows, Format-String-Attacken
    - Command-Injection, z.B. SQL-Injection

Sicherheitsprotokoll TLS/SSL

- SSL/TLS (Transport Layer Security) sichert die **Vertraulichkeit** der Kommunikation von Client/Server-Applikationen durch kryptographische Methoden.
  - Kein Abhören, Manipulieren oder Fälschen von Nachrichten möglich (**Verschlüsselung**, **Integritätsprüfung** der übertragenen Daten).
  - **Authentifizierung** des Servers und (optional) des Clients möglich mittels Public Key-Kryptographie (**X.509-Zertifikate**).
- **Transport Layer Security** Protokoll (TLS version 1.0), proposed Internet Standard (RFC 2246)
- Weiterentwickelt von Netscape **Secure Sockets Layer** (SSL) Protokoll Version 3.0

Sicherheitsprotokoll TLS/SSL

- Läuft über einem verlässlichen Transportprotokoll: z.B. über TCP.
- Bietet Kommunikationssicherheit unabhängig vom Anwendungsprotokoll.
- Jedes (verbindungsorientierte) higher-level Protokoll kann transparent über TLS gesichert werden: HTTP, LDAP, IMAP, ...



Sicherheitsprotokoll TLS/SSL

- Zwei Subprotokolle: **TLS Record Protocol** und **TLS Handshake Protocol**
  - **TLS Handshake Protocol** etabliert Sicherheits-kontext zwischen Kommunikationspartnern:
    - Authentifizierung des Servers gegenüber dem Client
    - (optional) Authentifizierung des Clients gegenüber dem Server
    - Aushandlung von Sicherheitsparametern (kryptographische Algorithmen etc.)
    - Benutzt Public-key-Kryptografie um shared secrets zu generieren.

TLS Handshake Protocol

Client		**Server**
ClientHello	-------->	
		ServerHello
		Certificate*
		ServerKeyExchange*
		CertificateRequest*
	<--------	ServerHelloDone
Certificate*		
ClientKeyExchange		
CertificateVerify*		
[ChangeCipherSpec]		
Finished	-------->	
		[ChangeCipherSpec]
	<--------	Finished
Application Data	<------->	Application Data

Sicheres WWW

- **HTTPS** = HTTP über SSL/TLS
- Server-Zertifikat bewirkt Verschlüsselung und Authentifizierung des Web-Servers.
- Optional auch Authentifizierung des Clients (Browser) durch Zertifikat.
  - Ersatz für schwächere Authentifizierung durch Passwort.
- Untersützt von den meisten Browsern (Netscape Navigator, MSIE, ...) und Web-Servern.
- Weiterhin:
  - Angriffe auf Web-Anwendungen möglich.
  - Kein erhöhter Schutz vor Angriffen auf Client.

Man-in-the-middle-Angriff

- Umleiten der Kommunikation über Rechner des Angreifers, z.B. mittels
  - DNS-Spoofing,
  - ARP-Spoofing im LAN oder
  - Proxy-Konfiguration des Browsers.
- Rechner des Angreifers fungiert als SSL-Server und SSL-Client für Zugriff auf angefragte Web-Site.
- SSL-Tunnel ist am Man-in-the-middle-Rechner unterbrochen.
- Mitlesen und Manipulation der Kommunikation ist möglich.

Man-in-the-middle-Angriff

- SSL-Protokoll kann dies erkennen, aber:
  - Ungenügende Fehlermeldungen des Browsers.
  - Mangelndes Anwenderbewusstsein.
    - Warnungen werden weggeklickt.
  - Public-key-Kryptographie ist kein Allgemeinwissen!
  - Mangelnder Schutz des Zertifikatsspeichers im Browser.
- Beispiel:
  - ISP verteilt Browser mit verändertem Zertifikatsspeicher und Proxykonfiguration des ISP.
  - Proxy generiert on-the-fly Zertifikate für die angefragten Web-Sites.
  - SSL-Absicherung ist am Proxy unterbrochen.

Beispiele für SSL-Nutzung

- Absicherung (Vertraulichkeit, Integrität, Authentizität) i.d.R. **Ende-zu-Ende**:
  - Http, telnet, ftp, ldap (s.u.) ...
  - https auch Ende-zu-Ende über Proxies mittels CONNECT-Methode
  - imap, pop3 (je nach Sichtweise)
- Absicherung i.d.R. nur über **Teilstrecken**:
  - SMTP, WebSphere MQ (message queuing von IBM), ...

Negotiation

- Bisher i.d.R. eigener Port für SSL-abgesichertes Protokoll.
  - Port 80 für http, Port 443 für https.
  - Port 389 für ldap, Port 636 für ldaps (deprecated, s.u.)
- Mittlerweile oft Protokollerweiterungen zur Aushandlung der Schutzfunktionen wie SSL.
  - SMTP: STARTTLS
  - Ldap: Start TLS Operation
  - Upgrading to TLS Within HTTP/1.1 (Proposed Standard)
  - ...

SMTP und SSL

STARTTLS

- Erweiterung des SMTP-Dienstes zur Nutzung von TLS zwischen SMTP-Server und -Client.
- Relaying auf Basis von Zertifikaten erlauben.
  - Absender muss sich zunächst authentifizieren.
- Ein- und ausgehende Verbindungen einschränken.
- Beachte:
  - Keine Ende-zu-Ende-Sicherheit,
  - Client ist nicht Autor der Email.

IP-Sicherheit

- Schwachstellen, Bedrohungen
  - Abhören der übertragenen Daten
  - Manipulation der übertragenen Daten
  - Vortäuschen falscher Identität durch Spoofing der IP-Adresse (oder des DNS-Namens)
  - Denial-of-Service-Angriffe
  - ...
- Protokolle höherer Schichten (TCP/UDP, Anwendungen) implementieren oft keine eigenen, weitergehenden Sicherheitsmechanismen.
  - Vertrauen auf IP-Adresse zur Client-Authentifizierung (Rechner, nicht Benutzer)
    - Z.B. r-Befehle mit rhosts-Sicherheit, NFS
  - Fehlende kryptographische Maßnahmen für Vertraulichkeit und Integrität der Kommunikation.
  - ...

IPsec

- Sicherheitsprotokoll in der Netzwerkschicht (Internetschicht) für IPv4 und IPv6
- Kryptographische Sicherheitsdienste unterstützen
  - Authentifizierung
  - Integrität
  - Zugangskontrolle
  - Vertraulichkeit
- Security Architecture for the Internet Protocol (RFC 2401)
  - Host-to-Host, Gateway-to-Gateway, Host-to-Gateway
- IP Encapsulating Security Payload (ESP) (RFC 2406)
  - Vertraulichkeit, Data Origin Authentication, Integrität, ...
  - Transport-Modus, Tunnel-Modus
- IP Authentication Header (AH) (RFC 2402)
  - Keine Vertraulichkeit
- Protokolle für Schlüsselmanagement (IKE, ISAKMP, ...)

Virtual Private Network

- Sichere Verbindung über unsichere Netze (z.B. Internet) mittels kryptographischer Sicherheitsdienste in den unteren Protokollschichten.
- IPsec im Tunnelmodus zwischen mehreren Security Gateways
  - Verbinden mehrerer privater Netze (Firmenlokationen) über öffentliche Netze (z.B. Internet)
- bzw. zwischen Host und Security Gateway
  - Remote-Access zum Unternehmensnetz von mobilen Rechnern/Heimarbeitsplätzen über öffentliche Netze (z.B. Internet)

Firewalls

- Generelles Konzept zur Absicherung von **Übergängen zwischen Netzwerken** mit unterschiedlichen Sicherheitsniveaus
- Gateways als **ein** Teil eines Sicherheits-konzepts
- In der Regel **mehrstufige** Firewall-Strukturen
- Weitere wichtige Komponenten
  - **Content Security** (Virenschutz, URL Blocking, ...)
  - Starke **Authentifizierung** für Remote Access
  - Schutz von Web-Servern und **E-Business-Systemen**

Firewalls

- **Anwendungsbereiche**
  - Internetzugang
  - Verbindungen zwischen unabhängigen Firmen
  - Anbindung von Heimarbeitsplätzen oder externen Vertriebsmitarbeitern an das Unternehmensnetz
    - Authentifizierung, Verschlüsselung, ggfs. Protokollierung
  - Verbindung von Filialnetzen über das Internet
    - Virtual Private Network (Verschlüsselung)
  - Trennung sensibler Netzbereiche vom restlichen Unternehmensnetz
    - Z.B. Forschung und Entwicklung, Personal, ...

Firewalls

- Zusammenspiel mit **internen Servern**
  - Nameserver
    - internes und externes DNS
  - Management-Systeme, Log-Hosts
  - Mail-Server
  - WWW-Proxies (Caching, Virenscanning, ...)
  - News-Server
  - E-Business-Systeme
    - Extern erreichbarer Web-Server
    - Interne Anwendungen und Datenbanken
  - ...

Firewalls

- **Grundkomponenten**
  - IP-Filter
  - Dynamische Filter
  - TCP-/UDP-Relays
  - Application Gateways/Proxies

IP-Filter

- Filtern jedes einzelnen Pakets
  - unabhängig von vorherigen Paketen
- In der Regel auf Basis der IP-Header-Felder
  - Protokoll (ICMP, TCP, UDP, ...)
  - Quell-IP-Adresse
  - Ziel-IP-Adresse
  - Quell-Port (bei TCP-/UDP-Paketen)
  - Ziel-Port (bei TCP-/UDP-Paketen)
  - Flags im TCP-Header
- Keine Pakete mit Source-Routing akzeptieren!
- Kein dynamisches Routing, keine ICMP-Redirects akzeptieren!

IP-Filter

- Meist in Hardware-Routern implementiert.
- Probleme:
  - Z.B. FTP (im aktiven Modus) benötigt weitere TCP-Verbindung für Datenübertragung von Server zu Client. → Großer Portrange für eingehende Verbindungen muss geöffnet werden.
  - Durch Manipulation von IP-Fragmenten können einige IP-Filter umgangen werden.
  - Fehlende Kenntnis der Anwendungsprotokolle, insbesondere des Status. → Kein Schutz vor anwendungsgetriebenen Angriffen.

Beispiel einer Access Control List eines Cisco-Routers

! HTTP proxy outbound
access-list 101 permit tcp host proxy1 gt 1023 any eq www
access-list 101 permit tcp host proxy1 gt 1023 any eq ftp
access-list 101 permit tcp host proxy1 gt 1023 any eq ftp-data *established*
! default rule
access-list 101 deny ip any any log
! HTTP proxy inbound
access-list 102 permit tcp any eq www host proxy1 gt 1023 *established*
access-list 102 permit tcp any eq ftp host proxy1 gt 1023 *established*
access-list 102 permit tcp any eq ftp-data host proxy1 gt 1023
! default rule
access-list 102 deny ip any any log
! Interface bindings
! Internal interface
interface ethernet 0
ip access-group 101 in
! External interface
interface serial 0
ip access-group 102 in

Dynamische Filter

- Status jeder Verbindung wird in einer Tabelle gespeichert („Stateful Inspection“).
- Antwortpakete werden der Verbindung zugeordnet.
- Z.B. FireWall-1 von CheckPoint

TCP-/UDP-Relays

- Verbindungsweiterleitung über ein dual-homed Gateway (ohne IP-Forwarding)
- Flexibler: Socks Version 5
- In der Regel Änderungen am Client nötig.

Application Gateways/Proxies

- Arbeiten vollständig auf **Anwendungsebene**.
- Nehmen Verbindungen für ein spezielles Protokoll entgegen, verarbeiten Daten auf Anwendungsebene und leiten diese weiter.
- Z.B. für Email (SMTP), WWW (HTTP), FTP
- Vollständige Trennung der Kommunikations-verbindungen zwischen internem und externem Netz.
- Semantik des Applikationsprotokolls bekannt.

→ Filterung auf Applikationsebene möglich.

→ Z.B. nur Download, kein Upload via FTP aus internem Netz.

- Hoher Aufwand, hohe Performance-Anforderungen.

Authentifizierung

- Kapitel 5: Authentifizierung
  - Passwörter
  - Token
  - Biometrische Verfahren
  - Kerberos

Begriffe

- Identifizierung
  - Wer ist mein Kommunikationspartner?
- Authentifizierung
  - Ist mein Kommunikationspartner tatsächlich der, der er zu sein vorgibt?
- Autorisierung
  - Welche Berechtigungen räume ich meinem Kommunikationspartner auf meinem System ein?

Authentifikationsverfahren

- Wissen
  - Passworte, PINs
- Besitz
  - SmartCard, EC-Card, SIM-Card
- Biometrische Merkmale
  - Fingerabdruck, Iris (Augenhintergrund), Gesicht
- Sicherheitskritische Anwendungen werden durch eine **Kombination** von Verfahren abgesichert (2-Faktor-Authentifizierung):
- Z.B. Besitz und Wissen:
  - Geldautomat, Handy

Klassifikation

- Schwache Authentifizierung
  - Passwörter
- Starke Authentifizierung
  - Einmal-Passwörter
  - Challenge-Response-Verfahren
- Zero-Knowledge Authentifizierungsverfahren
  - Neue Technik

Passwortverfahren (1)

- Benutzer oder System gibt Passwort mit einer bestimmten Gültigkeitsdauer vor.
- Vorteile:
  - Weit verbreitet.
  - Einfach und kostengünstig zu realisieren.
- Probleme:
  - Wahl der Passworte durch Benutzer sehr schlecht, durchschnittlich mehr als 80% leicht zu knacken.
  - Systemgenerierte Passwörter: Schwer zu merken.
  - Speicherung, Verwaltung.
  - Social Engineering: „Gib mir mal kurz dein Passwort.“
  - Speicherung zwar in verschlüsselter Form, aber in allgemein lesbarer Datei: z.B. /etc/passwd.
  - Ungeschützte Übertragung über Netze zum Server.

Passwortverfahren (2)

- Einmal-Passwörter
- Sehr sicher, da Passwort nur einmal verwendet.
  - 1. Möglichkeit: Vor der Übertragung werden lange Passwortlisten vereinbart (z.B. TAN-Liste).
  - 2. Möglichkeit: Benutzer und System können Passwörter berechnen (z.B. S/Key).
- Z.B. S/Key-Funktionsweise
  - Anfangswert a (Seed), geheimer Schlüssel K, Länge der Passwortliste l, Hashfunktion H
  - $P_1 = H(a, K)$, $P_2 = H(P_1,K)$, ... , $P_l = H(P_{l-1},K)$
  - Benutzer und System müssen a, K, l am Anfang vereinbaren und natürlich geheim halten.

Attacken auf Authentifizierung

- Offline Attacke:
  - Der Angreifer bringt sich in den Besitz von Informationen (Kopieren der Passwortdatei, Schnüffeln von Authentifizierungsdaten im Netz) und wertet diese „in aller Ruhe“ in seinem Umfeld aus.
  - Gegenmaßnahmen: Schutz der Passwortdateien durch Verschlüsselung und restriktive Autorisierung, keine Klartextübertragung von Authentifizierungsdaten.
- Online Attacke:
  - Der Angreifer versucht z.B. durch Probieren die Authentifizierung direkt zu erreichen.
  - Gegenmaßnahmen: Beschränkte Eingabeversuche, Verzögerung zwischen Eingabeversuchen, Zwang zu Anwesenheit.

Challenge Response (1)

- Challenge-Response-Verfahren (symmetr.)
  - Benutzer und Server vereinbaren geheimen Schlüssel K und Funktion F (z.B. Hash-Funktion).
  - Server übermittelt eine Zufallszahl z an Benutzer und berechnet F(z,K).
  - Benutzer berechnet F(z,K) und übermittelt Ergebnis an den Server, der Gleichheit überprüfen kann.
- Vorteile:
  - Passwort wird nie übertragen.
  - K kann auf Benutzerseite z.B. in einem Token (Chipkarte o.ä.) gespeichert werden.

Challenge Response (2)

- Challenge-Response-Verfahren (asymmetr.)
  - Server kennt öffentlichen Schlüssel P des Benutzers.
  - Server übermittelt Zufallswert z an Benutzer.
  - Benutzer signiert z und übermittelt Signatur.
  - Server kann mit öffentlichem Schlüssel Signatur überprüfen.
- Vorteil:
  - Passwort wird nie übertragen.
  - Existierende PKI kann von vielen Anwendungen zur Authentifizierung genutzt werden.
  - Ebenfalls Benutzung von Token möglich.

Zero-Knowledge-Protokolle

- Das Problem herkömmlicher Challenge-Response-Verfahren:
  - Einem Angreifer wird z.B. mit jeder abgehörten Authentifizierung etwas mehr Information über den geheimen Schlüssel bekannt.
- Zero-Knowledge-Protokolle:
  - Es ist nicht möglich, aus der Durchführung einer Authentifizierung etwas über die geheimen Protokollparameter zu erfahren.
- Beispiel:
  - Fiat-Shamir-Protokoll
  - Ähnlichkeit mit RSA-Verfahren (Ausnutzung des Faktorisierungsproblems), aber geheimer und öffentlicher Schlüssel werden ständig neu erzeugt.

Token (1)

- Kryptographische Schlüssel sind für Benutzer einfacher zu handhaben, wenn sie in abgeschlossener, kleiner Hardware transportiert werden können.
- Besitz (Token) und Wissen (PIN zum Bedienen des Token) können zur 2-Faktor-Authentisierung kombiniert werden.
- Z.B. Challenge-Response-Verfahren mit taschenrechnerähnlichem Token:
  - Benutzer schaltet mit PIN Token ein.
  - Benutzer gibt Server-Challenge in Token ein.
  - Token berechnet mit gespeichertem Schlüssel Antwort.
  - Benutzer schickt Antwort zum Server.

Token (2)

- Token ohne Challenge-Response:
- Z.B. SecureID-Cards:
  - Berechnung eines Einmal-Passworts aus einem geheimen Schlüssel, der aktuellen Zeit und der Benutzer-PIN.
  - Server kennt geheimen Schlüssel und kann Rechnung nachvollziehen.
  - Änderung des Passworts z.B. alle 60 sec.
  - Begrenzte Lebensdauer der Karten: Abschaltung nach 3 Jahren.

Token (3)

- Chip-Karten:
  - Speicher-Karte:
    - Nicht-flüchtiger Speicher, keine CPU, sehr billig.
    - Z.B. Telefonkarte, Krankenversicherungskarte.
  - SmartCard:
    - Chip mit CPU, ROM, RAM und EEPROM.
    - ROM speichert Betriebssystem, Kryptoverfahren, PIN-Prüfungsalgorithmus etc.
    - EEPROM speichert Schlüssel, PIN, Kontonummer etc.
    - Z.B. Geldkarte, SIM-Karte für Handy.
- USB-Token
  - Gleiche Funktionen, andere physikalische Schnittstelle.

SmartCard-Sicherheitsdienste

- Authentifizierung
  - Benutzer authentifiziert sich mit PIN gegenüber Karte.
  - Karte authentifiziert sich mit Challenge-Response gegenüber Kartenleser.
- Vertraulichkeit
  - Verschlüsselung zwischen Kartenleser und SmartCard.
- Integrität
  - MAC-Berechnung zwischen Kartenleser und SmartCard.
- Verbindlichkeit
  - Erstellen digitaler Signaturen.

Beispiel GSM: SIM-Karte

- SIM-Karte enthält u.a.
  - Teilnehmerkennung,
  - Authentifizierungsschlüssel,
  - PIN.
- Benutzer authentifiziert sich gegenüber Karte mit PIN.
- SIM-Karte authentifiziert sich mit symmetr. Challenge-Response gegenüber Netz.
- Aushandlung eines Sitzungsschlüssels zur vertraulichen Kommunikation.
- Problem: Das Netz authentifiziert sich nicht!

Biometrische Verfahren

- Authentifizierung anhand der Wiedererkennung unverwechselbarer und unveräußerlicher Merkmale
  - Z.B. Iris-Erkennung, Gesichtserkennung, Fingerabdruck.
- Notwendig: Speicherung von Referenzmustern
  - Abweichung vom Original unvermeidlich.
  - Bei der Überprüfung müssen Toleranzwerte festgelegt werden.
- Fehlertypen
  - Abweisung berechtigter Benutzer: Akzeptanzproblem.
  - Unberechtigter wird authentifiziert: Sicherheitsproblem.
- Bisher noch kein Erfolg auf breiter Front, Zukunft fraglich.
  - Unausgereifte Technik, keine Akzeptanz, inakzeptable Fehlerraten, Eingriff in Persönlichkeitsrechte, ...

Einwahlauth.: PAP vs. CHAP

- PAP = Password Authentication Protocol
- Verwendet bei Einwahl via PPP.
- Kennung und Passwort im Klartext.
- Keine Unterbindung bei Missbrauch möglich.
- Initiative ergreift der Client.

- CHAP = Challenge Handshake Authentication Protocol
- Verwendet bei Einwahl via PPP.
- Server sendet Challenge.
- Keine Passwortübertragung.

Einwahl: RADIUS / TACACS

- PAP und CHAP dienen nur der Authentifizierung der PPP-Verbindung.
- Eine Datenbank mit Kennung/Passwort ist vorzuhalten.
- Authentifizierungsdatenbanken sind aber in Netzwerken normalerweise bereits vorhanden.
- Abhilfe: RADIUS(Remote Dial-In User Service)-Protokoll oder TACACS-Protokoll bietet Verbindung zwischen Authentifizierungsserver und Einwahlauthentifizierung.

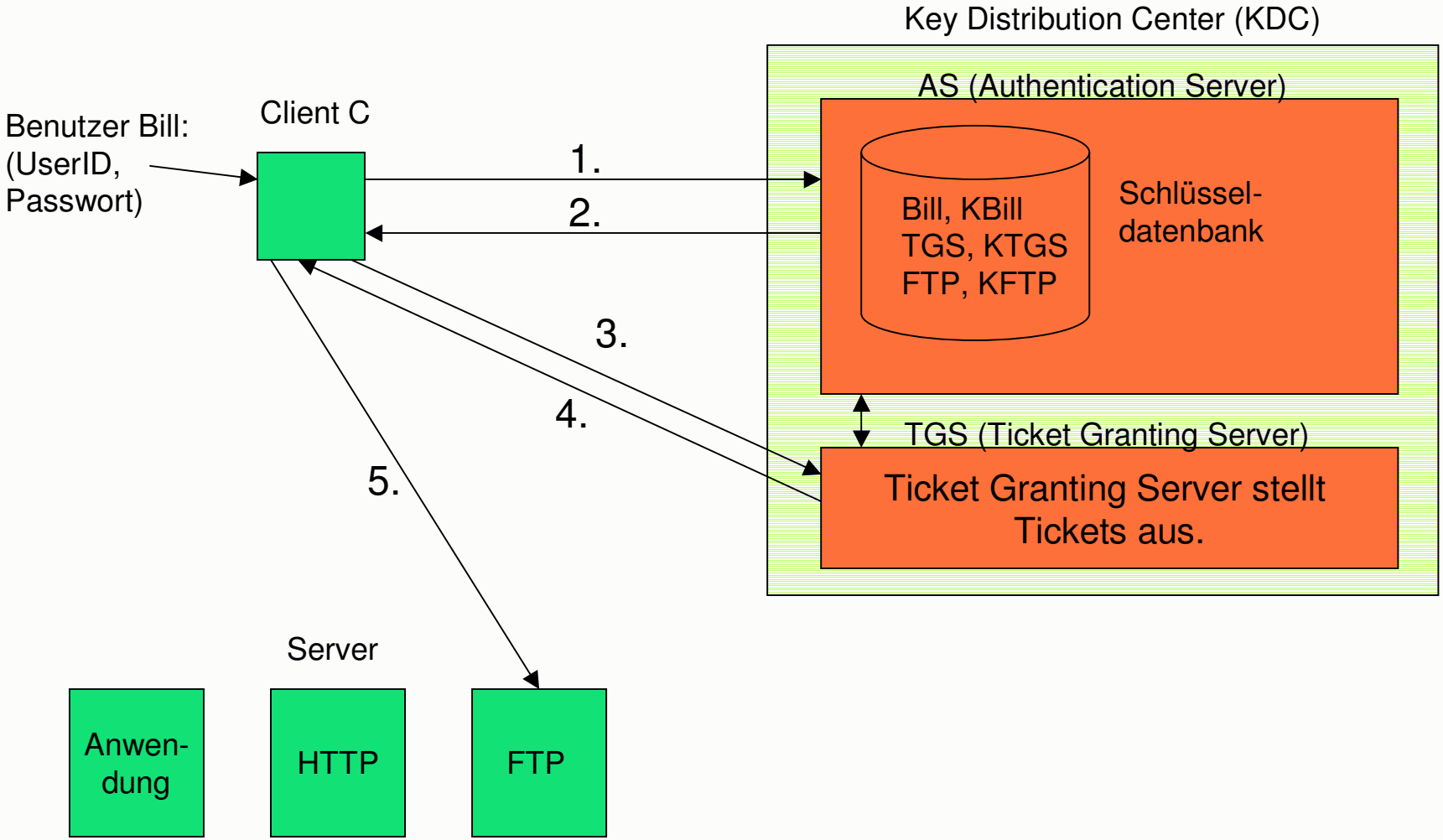
Kerberos

- Netzwerkauthentifizierungsprotokoll
- Kerberos-Server hält alle Schlüssel aller angeschlossenen User, Geräte und Dienste (Principals) im Netzwerk.
- Wenn ein Principal einen Dienst nutzen will, authentifiziert er sich mit Passwort gegenüber dem Kerberos-Server, der ein zeitlich begrenztes „Ticket“ zur Nutzung des Dienstes für den Principal ausstellt.
- Möglichkeit für Single-Sign-On.
- In vielen Betriebssystemen implementiert (Unix, Windows).
- Ermöglicht Authentifizierung und Sitzungsschlüsselaustausch.

Kerberos Funktionsweise (1)

- Der Principal „Benutzer“ meldet sich normal am Client mit UserID/Passwort an.
- Alles weitere für Benutzer und andere Principals transparent.
- Initialisierung: Vereinbarung eines geheimen Schlüssels K_A zwischen KDC und Principal A.
- Weitere Begriffe:
  - Authentikator für Client C: A_C bestehend aus C, Adresse von C und timestamp.
  - Nonce: Eindeutiger, nie zuvor verwendeter Identifikator, z.B. Zufallszahl.

Kerberos Funktionsweise (2)



Kerberos Funktionsweise (3)

Von	An	Nachrichten (Benutzer Bill will Zugriff auf FTP-Server)
1. Client	KDC	Bill, TGS, Nonce1
2. KDC	Client	$K_{Bill}(K_{Bill,TGS}, Nonce1), K_{TGS}(T_{Bill,TGS})$
3. Client	TGS	$K_{Bill,TGS}(A_{Bill}), K_{TGS}(T_{Bill,TGS})$, FTP, Nonce2
4. TGS	Client	$K_{Bill,TGS}(K_{Bill,FTP}, Nonce2)$, $K_{FTP}(T_{Bill,FTP})$
5. Client	FTP-Server	$K_{Bill,FTP}(A_{Bill}), K_{FTP}(T_{Bill,FTP})$

Sicherheit von Kerberos

- Kerberos erhöht die Sicherheit in Client-Server-Umgebungen erheblich, aber:
- Probleme (Auswahl):
  - Verwaltung von Sitzungsschlüsseln auf Clientsystemen.
  - Beteiligte Rechner müssen sich zeitlich synchronisieren.
  - Es werden nur Passworte zur Authentifizierung am Client eingesetzt.

Drahtlose Kommunikation

- Kapitel 6: Drahtlose Kommunikation
  - Wireless LAN (WLAN)
  - Bluetooth

Sicherheit von WLAN

- Verbreitung geschieht häufig unkontrolliert in Unternehmensnetzen.
  - Schnelle Erweiterung von kabelgebundenen LANs.
  - Günstige, leicht verfügbare Geräte.
- Die physikalische Zugangskontrolle zum LAN entfällt!
  - Reichweite bis zu 300m im Freien.
  - Abhören auch noch in 10 km Entfernung möglich.
- Standard für WLAN (IEEE 802.11b) enthält massive Sicherheitsprobleme.

WEP

- Sicherheitsprotokoll Wired Equivalent Privacy (WEP) bietet keine Wired Equivalent Privacy.
  - Schlüsselmanagement (kein Schlüsselaustauschprotokoll)
  - 40-bit- und 104-bit-Verschlüsselung (Angriffe dauern in der Regel nur Stunden.)
  - Schwachstellen in der Nutzung des Verschlüsselungsalgorithmus RC4 (U.a. bietet 104-bit WEP keine wesentliche Verbesserung gegenüber 40-bit.)

WEP Authentifizierung

- Zwei „Authentifizierungsschemata“ sind möglich:
  - Open System:
    - Keinerlei Überprüfung, jeder mobile Teilnehmer wird als Nutzer zugelassen. → Keine Authentifizierung.
    - Default Einstellung!
  - Shared Key:
    - Nutzung des WEP-Schlüssels mit Challenge Response.
- Eine Authentifizierung des WLAN gegenüber dem Client ist nicht vorgesehen.
- Der Zugriff kann auf bestimmte MAC-Adressen beschränkt werden.
  - Kein wirklicher Schutz, da MAC-Spoofing einfach ist.

WEP Integrität

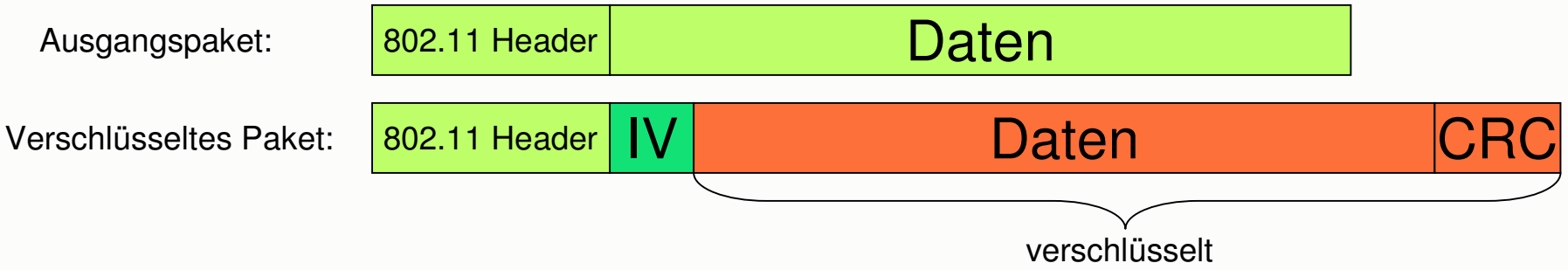
- Sender berechnet mittels CRC-32 Verfahren eine Prüfsumme und hängt sie an die Nachricht (32 Bit).
- CRC-32 ist effizientes Verfahren zum Checken von Übertragungsfehlern.
- Keine starke Hash-Funktion!
- Keine Benutzung des WEP-Schlüssels.
- Praktisch keine Integritätssicherung!

WEP Vertraulichkeit

- Verwendung eines geheimen Schlüssels K (40 oder 104 Bit) und des RC4 Verfahrens.
- Damit nicht jedes Paket mit demselben Schlüssel verschlüsselt wird, wird ein zufälliger Initialisierungsvektor (IV) von 24 Bit Länge erzeugt.
- Verschlüsselt wird nicht allein mit K, sondern mit

- → IV muss im Klartext mitgesendet werden.

WEP Vertraulichkeit



- IV nur 24 Bit lang. → Bei stark frequentierten Access Points braucht man nur wenige Stunden, bis sich IV‘s wiederholen.
- IV Bestimmung ist in vielen Implementierungen nicht zufällig.
- Eine Neuwahl des IV mit jedem Paket ist nicht vorgeschrieben.

Weitere WEP Probleme

- Schlüssel müssen manuell in jedem Gerät konfiguriert werden.
  - Schlüsselwechsel bedeutet hohen Aufwand.
  - Schlüssel müssen vielen bekannt sein.
- Wegen kurzem IV bringt die Vergrößerung des Schlüssels von 40 auf 104 Bit fast nichts.
- FAZIT: Auch ohne das direkte Brechen des RC4 Algorithmus bietet WEP wegen der indiskutablen Implementierung von kryptografischen Funktionen praktisch keine Sicherheit:
- WLAN ist ein quasi offenes Netz!

Sicherheit von WLAN

- Hackerwerkzeuge (Netstumbler, AirSnort, ...) leicht verfügbar.
- War Driving
- Endgeräte sind im WLAN angreifbar.
- Denial-of-Service-Angriffe sind möglich (Jammer).

Neue Entwicklungen

- WPA - WiFi Protected Access sieht gegenseitige Authentifizierung vor, standardisiert aber die Methode nicht.
  - Geräte-Inkompatibilitäten zwischen Herstellern.
  - WPA automatisiert regelmäßigen Schlüsselwechsel.
  - WPA ist Zwischenlösung bis zur Ratifizierung von 802.11i.
- Neuer Sicherheitsstandard 802.11i (u.a. mit AES)

Bedrohungen

- Bedrohungen
  - Mithören der Kommunikation
  - Manipulation der übertragenen und verarbeiteten Daten
  - Unautorisiertes Eindringen ins Unternehmensnetz
  - Angriffe auf die Verfügbarkeit von Anwendungen und Systemen
- Gleiche Bedrohungen durch WLAN wie durch Fernzugänge über das Internet.
  - Bedeutet geringere Reichweite ein geringeres Risiko?
- Fazit: Funknetze (WLAN) wie öffentliche Netze (z.B. Internet) behandeln.

Schutzmechanismen

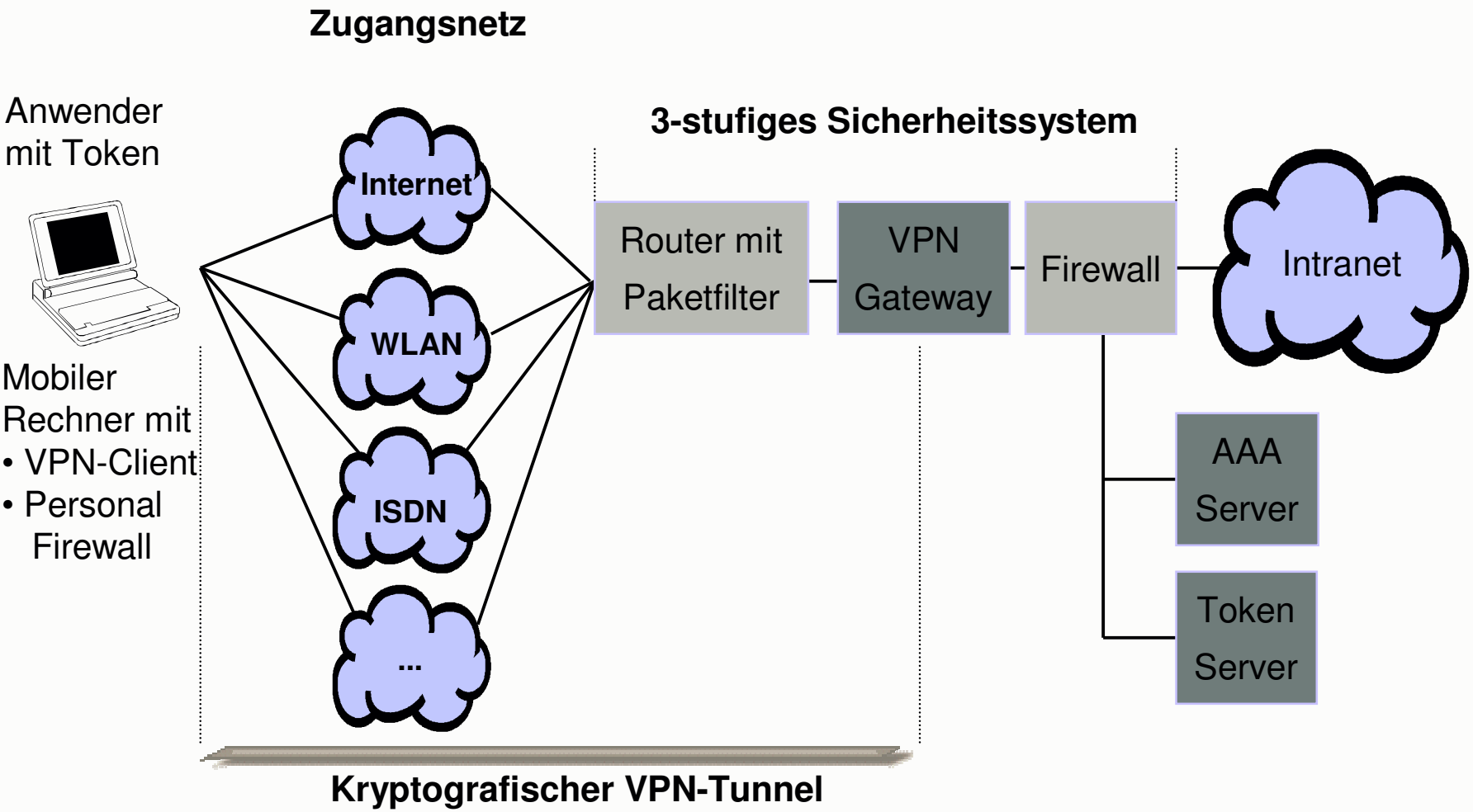
- Lösungsvarianten
  - Verschiede Grundschutzmaßnahmen (Sicherheitskonfiguration der Access Points, WEP aktivieren, Schlüssel regelmäßig wechseln (aufwändig), MAC-Adressenverwaltung (aufwändig), ...)
    - Für Einsatz in größeren Unternehmen nicht ausreichend.
  - Kombination mit anderen Sicherheitstechnologien (analog zu Fernzugängen über das Internet)
    - Für Einsatz in größeren Unternehmen geeignet.
- Lösungsansatz
  - Behandlung von WLAN wie Remote-Access über öffentliche Netze.

Sicherheitsanforderungen

Im Umfeld größerer Unternehmen:

- Starke **Authentifizierung** des Benutzers über Hardware-Token nach dem Prinzip Wissen (PIN) und Besitz (Token),
- starke **Verschlüsselung** (z.B. IPsec-basierter VPN-Tunnel),
- **Netzwerk-Zugriffsschutz** auf interne Ressourcen,
- Integration der Benutzerverwaltung in bestehende Verfahren,
- Schutz des Rechners und des internen Netzes vor den Gefahren des Internet (**Personal Firewall**),
- weitere Schutzmaßnahmen für mobile Rechner (Virenschutz, Betriebssystemhärtung, Dateiverschlüsselung),
- zentrale Sicherheitsadministration des Clients (VPN, Personal Firewall, Virenschutz, ...),
- Protokollierung.

Remote-Access mittels VPN



Bluetooth

- PAN (Personal Area Network)
  - Kommunikation eines bzw. weniger Nutzer im Umkreis von bis zu 10 m.
- Viele Nutzungsszenarien
  - Von der Kopplung von Peripheriegeräten bis zur Kopplung von PC‘s.
- Unterstützung von Ad-hoc-Verbindungen
  - Pico Netze mit bis zu 8 Teilnehmern.
- Unterstützung von Sprach- und Datenkanälen

Bluetooth Sicherheit

- Maßnahmen sind auf der Verbindungsebene definiert (Link Level).
- Drei Security-Modi
  - non-secure
    - Gerät ignoriert Sicherheitsdienste.
  - service level enforced security
    - Sicherheit auf Dienstebene nach Aufbau einer Verbindung möglich.
  - link level enforced security
    - Authentifizierung und Verschlüsselung auf der Verbindungsebene.
    - Bei erstmaliger Kommunikation „Pairing“ (Aushandeln eines Verbindungsschlüssels mit einer PIN) nötig.

Bluetooth Sicherheit

- Eine Pflicht zur Implementierung der Sicherheitsfunktionalitäten ist nicht vorhanden.
- Bei Pairing sollten nur lange und zufällige PINs zum Einsatz kommen.
- Abhören des Pairing mit zu kurzer PIN kann mit Brute-force-Attacken ausgenutzt werden.
- Trotzdem: Alles in allem ist bei Bluetooth die Implementierung von Sicherheits-funktionalitäten erheblich besser gelungen als bei WLAN.