INFORMATIONSTECHNIK
Ausstellungsstrasse 70
CH-8090 Zürich
Homepage www.tbz.ch
Telefon 044 446 96 60
Telefax 044 446 96 66
E-Mail admin.it@tbz.zh.ch

Modul 150

E-Business-Applikationen anpassen

Gerd Gesell © Mai 2010 Version: 1.5

Ersatzheft: 10,- sFr.

Inhaltsverzeichnis

Präambel

Das Originalskript für das Modul 150 wurde im Jahr 2004 durch Philipp Gressly mit dem Textverarbeitungssystem LaTeX 2_ε verfasst. Dies habe ich weitergeführt und die neuen Versionen ebenfalls damit erstellt.
Verwendet habe ich TeXnicCenter[1].

[1] http://www.texniccenter.org

1 Modulidentifikation

Modulnummer	150
Titel	E-Business-Applikationen anpassen
Kompetenz	E-Business-Applikationen gemäss Vorgabe und unter Beachtung der Sicherheitsvorschriften und technischer Rahmenbedingungen anpassen.
Handlungsziele	1. Technik der Applikation, Transaktionskonzept, Applikationsumgebung und Rahmenbedingungen (Sicherheit, Performance, Verfügbarkeit, Transaktionsvolumen, usw.) erfassen 2. Vorgabe analysieren, clientseitige, serverseitige und datenbankseitige Umsetzung ableiten und dokumentieren. 3. Vorhaben auf Sicherheit und Schutzwürdigkeit der Informationen über alle Ebenen (Client, Webserver, Applikationsserver und Datenbankserver) überprüfen und dokumentieren. 4. Konzept für die Anpassung der E-Business-Applikation erarbeiten, Test- und Abnahme-Konzept erstellen und Implementierung planen und dokumentieren. 5. Anpassungen auf GUI-, Applikations- und Datenbankebene realisieren bzw. programmieren und dokumentieren. 6. Umsetzung gemäss Testkonzept auf Funktionalität, Fehlerfreiheit, Performance, Schutz und Sicherheit überprüfen und Testergebnisse im Testprotokoll festhalten. Allenfalls erforderliche Korrekturen durchführen.
Kompetenzfeld	Web Engineering
Objekt	• Online Shop Lösung. • Produktverzeichnis mit online Bestellwesen ergänzen.
Niveau	4
Voraussetzungen	• E-Commerce Prozesse strukturieren und interpretieren • Erfahrung mit client- und serverseitigen Skriptsprachen • Web-Applikationen realisieren • Anforderungen an den Datenschutz, die Sicherheit und die Verfügbarkeit im • E-Commerce Segment definieren
Anzahl Lektionen	40
Anerkennung	Eidg. Fähigkeitszeugnis Informatiker/Informatikerin
Modulversion	1.1
MBK Release	R3
Harmonisiert am	19.09.2006

2 Einleitung und Rahmenbedingungen

2.1 Begriffsdefinitionen - Modul 150

Mit dem Präfix „E-“ versehen, geistern in der Informatik viele Begriffe herum. Auch in der Modulidentifikation zu diesem Modul werden die Begriffe *E-Business* und *E-Commerce* parallel verwendet. Je nach verwendeter Literatur, werden zwischen diesen beiden Begriffen aber wesentliche Unterschiede gemacht. In diesem Skript werden wir uns aus mühsamen Begriffsdefinitionen möglichst heraus halten und einen Überblick über Techniken und Grundlagen geben, die bei der Anpassung und beim Entwurf von Webapplikationen eine Rolle spielen. Wenn hier im Skript der Begriff *E-Business* verwendet wird, bezeichnet er Applikationen, die mindestens die Komponenten Produkteauswahl, Zahlungsverkehr und Warentransport (elektronisch oder physikalisch) enthalten.

Soziale Plattformen, Foren und so weiter gehören nicht zu dieser Applikationsgruppe. Die meisten hier betrachteten Techniken werden auch bei diesen Anwendungen eingesetzt.

Sie sollten in diesem Modul einen Überblick über Themen bekommen, die im Bereich E-Business wichtig sind. Die können in 40 Lektionen zwar nicht vollständig bearbeitet werden, aber eine Sensibilisierung auf mögliche Probleme und der Erwerb von Grundkenntnissen mit zusätzlicher selbstständiger Vertiefung sind sicher möglich.

2.2 Projektmanagement

Die Abwicklung von Webprojekten und die Realisierung von E-Business Lösungen erfordert ein professionelles Projektmanagement. Welche Methoden dabei eingesetzt werden, ist von den beteiligten Unternehmen und von der Anwendung abhängig und nicht Bestandteil dieses Moduls. Lediglich der Begriff *Change Management* sei hier noch erwähnt, auf den wir später kurz eingehen werden.

2.3 Partnerrollen bei Webapplikationen

Auch wenn die Begriffsdefinitionen hier nicht im Vordergrund stehen, müssen wir eine Reihe von Merkmalen kennen, da sie die Applikation und auch die eingesetzte Technik stark beeinflussen. Wir müssen uns klar machen, welche Unterschiede in Bezug auf die Geschäftspartner bei Webapplikationen bestehen und in welchem Gebiet unsere Applikation eingesetzt wird. Das hat Konsequenzen auf den Entwurf, die anzuwendenden Gesetze, die Technik und die Gestaltung der Applikation.
Anmerkung bezüglich der Partnerrolle „C“: In der Literatur und auch im Internet finden wir die Bezeichnungen *customer* und *consumer*. In Verbindung mit der Rolle „A“ für *administration* bedeutet es dann *citizen* also Bürger.

2.3.1 B2C - Business to Customer/Consumer

Diese Geschäftsbeziehung fällt einem vermutlich als erste mögliche Beziehungsvariante ein. Sie entspricht dem klassischen Ladengeschäft, das seine Waren an

Endkunden verkauft. In der elektronischen Variante ist dazu kein physikalischer „Laden“ mehr nötig, aber viele der gesetzlichen Bestimmungen zum Thema Konsumentenschutz, sind mittlerweile auch im Internet-Handel gültig. Etwas komplizierter wird es durch die Globalisierung via Internet. Wenn zwischen den Ländern des Verkäufers und des Käufers Gesetze und Vereinbarungen bestehen, ist die Unsicherheit zwar limitiert, aber Zölle, Einfuhrbeschränkungen auf bestimmte Güter, Versandmodalitäten und eventuell auch sprachliche Schwierigkeiten sorgen für ausreichend Möglichkeiten einen Geschäftsvorgang kompliziert werden zu lassen. Was die juristischen Anforderungen zum Thema Konsumentenschutz betrifft, sind hier besonders das Rückgaberecht und die Auszeichnungspflicht (Angabe des Warenpreises inklusive Mehrwertsteuer) zu nennen. Diese Kontellation der Geschäftspartner entspricht auch meistens den von uns untersuchten Webshop-Anwendungen.

2.3.2 B2B - Business to Business

Hierbei handelt es sich um die Beziehung zweier Geschäftspartner. Beide Partner sind sich im Normalfall bekannt. Zumindest muss eine Überprüfung der Autorisierung erfolgen, um sicher zu stellen, dass es sich bei beiden Partnern um Firmen handelt. Vor unberechtigten Zugriffen durch Endkunden muss die Anwendung geschützt werden. Bei Verkaufsgeschäften bestehen keine Rücknahmepflichten und die Preisauszeichnung erfolgt ohne Angabe der Mehrwertsteuer. Ein privater Kunde hat andere Anforderungen und Erwartungen an eine Applikation als ein Lieferant oder Zwischenhändler. Zumeist wird diesen mehr Können in Bezug auf Usability und Ziele der Applikation zugemutet als es bei anonymen Endkunden möglich ist. So kann die Schnittstelle durchaus einmal aus einem autorisierten FTP[2]-Download einer Datenbank bestehen.

2.3.3 C2C - Customer to Customer

Das wohl erfolgreichste Beispiel für diese Kundenbeziehung ist EBay™. Ursprünglich ging es hier um einen Gebrauchtwarenhandel, der zu Beginn ausschliesslich zwischen Privatpersonen durchgeführt wurde. Aktuell tummeln sich aber auch jede Menge Händler auf der Plattform und bieten Neuwaren an. Allerdings habe diese die Verpflichtung sich klar als solche zu kennzeichnen und die Bedingungen entsprechen dann einer B2C-Beziehung.
Findige (und zumeist auch windige) Juristenbüros haben diesen Umstand ausgenutzt und untersuchen die Anbieter nach ihren Artikel- und Angebotsprofilen. Wer eine alte CD-Sammlung auflöst, verkauft verschiedene CDs einmal und selten vielleicht zweimal. Wer aber die gleiche CD mehrfach verkauft muss sich eine gute Argumentation zurechtlegen, warum er sich nicht als Händler ausgibt und muss entsprechend mit Abmahnungen rechnen (siehe auch Kapitel 2.4.1 auf Seite 9).

2.3.4 A2C - Administration to A,B,C

Es gibt eine ganze Reihe von eBegriffen, die im Bereich von Behördenapplikationen verwendet werden. eGovernment, eAdministration, eJustice und eDemocracy

[2] File Transfer Protocol

sind einige davon. Interessant ist die derzeitige Diskussion in den Diskussionsforen der deutschen Wikipedia (zum Beispiel beim Begriff E-Government). Es wird wohl noch einige Zeit verstreichen, bis sich feste Begriffe mit eindeutigen Definitionen durchgesetzt haben. Generell geht es aber um Applikationen, die Bürgern, Firmen oder anderen Behörden angeboten werden, um den Kontakt auf elektronischem Weg zu ermöglichen. Das können stark sicherheitsrelevante Anwendungen wie Steuerabrechnungen, E-Voting (wählen und abstimmen über das Internet) und Ausschreibungen sein, aber auch einfache Applikationen, die den Gang zur Verwaltung ersparen.

2.3.5 B2E - Business to Employee

Anwendungen, die sich an die Angestellten einer Firma wenden, unterliegen ebenfalls Bedingungen zum Datenschutz und auch zur Datensicherheit. Besonders der Zugang von unautorisierten Personen ist hier ein Thema, denn sonst wäre es ja keine reine B2E-Applikation mehr. Diese Applikationen laufen meistens auf dem Intranet, da sie am Arbeitsplatz benötigt werden, sie können aber teilweise auch via Internet sinnvoll sein, wie zum Beispiel ein spezieller Webshop, der sich nur an Mitarbeiter richtet. Diese Kategorie ist in der Übersicht nicht vertreten.

	Administration	**Business**	**Consumer**
Administration	**A2A** Applikationen zwischen Behörden z. B. Interpol	**A2B** Applikationen für öffentliche Ausschreibungen;	**A2C**
Business	**B2A**	**B2B**	**B2C** Webshops; Anbieter sind gewerbsmässige Unternehmen
Consumer	**C2A**	**C2B** Angebotssuche potentielle Kunden schreiben ihre Projekte aus, Firmen bieten an	**C2C** Kleinanzeigen, Flohmärkte, alle privaten Handelsmöglichkeiten

Abbildung 1: Anwendungsbereiche des eBusiness

2.4 Juristische Grundlagen

Aufgrund der vermeintlichen (aber definitiv nicht realen) Anonymität im Internet und der automatischen Globalisierung von Webauftritten, wird das Internet von vielen als rechtsfreier Raum angesehen. Das kann aber spätestens dann zu einem bösen Erwachen führen, wenn man eine Abmahnung oder gar eine Klage erhält. Zumindest mit den wesentlichen rechtlichen Begriffen und Gepflogenheiten muss man sich als Webentwickler auskennen. Dabei ist es gleichgültig, ob es sich bei den erstellten Auftritten um professionelle oder um private Seiten handelt.

2.4.1 Abmahnungen

Bei Abmahnungen handelt es sich ursprünglich um Verfahrensvereinfachungen im Wettbewerbsrecht. Wer einen Verstoss im Wettbewerbsrecht feststellt, war damit nicht mehr gezwungen auf dem gerichtlichen Weg vorzugehen, sondern konnte aussergerichtlich direkt mit der anderen Partei Kontakt aufnehmen und den Fall lösen. Erst wenn sich die Partner nicht einigen können werden die Gerichte damit belastet. Bei offensichtlichen Verstössen und einsichtigen Konfliktparteien, können damit nicht nur die Gerichte entlastet werden, sondern auch die Folgekosten für die widerrechtlich handelnde Partei stark reduziert werden.
Soweit die grundsätzlich positive Idee dieses Rechtsmittels.
Leider wird dieses Instrument auch stark missbraucht. Es hat sich als ein einträgliches Geschäft erwiesen, im Internet nach kleinen Verstössen zu fahnden und die Urheber einfach abzumahnen, in der Hoffnung, dass diese - bei relativ geringen Abmahngebühren - den Weg des geringsten Widerstandes nehmen und bezahlen. Ein grosser Teil der Abmahngründe ist dabei juristisch nicht haltbar. Natürlich hat sich auch eine Gegenbewegung gebildet, die bei Bedarf Hilfe anbietet. Zum Beispiel `http://www.abmahnwelle.de`.

2.4.2 Impressum

Die Impressumspflicht wurde bereits in den Web-Modulen erwähnt. Problematisch ist die Unterscheidung zwischen EU- und Schweizer Recht. Noch problematischer sind dann Kombinationen: eine Schweizer Internetfirma hostet und entwirf den Auftritt für ein EU-Unternehmen und so weiter. Neben diesen juristischen Finessen ist es bei eBusiness-Applikationen aus Gründen der Transparenz und Offenheit wichtig, den Kunden nicht mit anonymen Partnern arbeiten zu lassen. Bei den meisten Applikationen ist ein Ziel trotz der vermeintlichen Anonymität des Internets eine Kundenbeziehung herzustellen und zu pflegen.

2.4.3 Disclaimer

Die juristische Relevanz von Disclaimern wird in der Praxis zumeist überschätzt. Grossfirmen haben gelegentlich am Ende ihrer Mails einen Hinweis auf die Vertraulichkeit des Inhalts und dass bei fehlgeleiteten Mails deren Inhalt nicht weitergegeben werden darf. Es ist halt ein Versuch, aber auch nicht mehr. Denn juristisch entspricht das Geschäftsbedingungen. Und diese sind vor einer Geschäftshandlung bekannt zugeben und zu akzeptieren. Korrekt, aber praktisch nicht machbar wäre es, zuerst den Disclaimer zu senden und - falls der Empfänger die Bedingungen

akzeptiert - erst anschliessend den Inhalt.
Ebenso unwirksam sind die meisten Disclaimer auf Webseiten. Es ist nicht sehr glaubwürdig auf andere Webseiten zu verweisen und sich gleichzeitig von diesen zu distanzieren. Wer in seriöser Absicht auf eine andere Webseite verlinkt, sollte deren Inhalt auch kennen. Wenn der Inhalt nach dem Einbau des Links verändert und eventuell juristisch bedenklich geworden ist, wird einem das kaum als Vergehen angerechnet werden. Allerdings muss man nach einem entsprechenden Hinweis darauf reagieren und seinen Eintrag anpassen.

2.4.4 Urheberrechtsschutz

Ein Rechtsgebiet, das auch noch im Bereich von eDistribution (siehe auch Kapitel 3.2.5 auf Seite 13) erwähnt wird, ist das Urheberrecht. Es bezieht sich generell auf alle Medien. Hier sei besonders auf die Situation im Bildbereich hingewiesen. Für die Gestaltung der Produktkataloge werden zur Illustration oft Produktfotos eingesetzt. Selbst wenn Sie diese „nur“ von der Herstellerseite herunterladen und verwenden, verstossen Sie gegen das Urheberrecht. Ausser die Fotos sind explizit zur freien Verwendung deklariert. Selbst wer Fotos kauft, muss sich über die Rechte seines Verkäufers informieren. Wenn später der Fotograf seine Rechte geltend macht, kann die Verantwortung zwar an den Verkäufer weitergegeben werden, nur hat man Pech, falls dieser Konkurs gegangen ist oder nicht mehr auffindbar ist. Daher verwenden viele Firmen für Ihre Webauftritte ausschliesslich Fotos, deren Urheberrecht sie selber haben. Die also durch einen Fotografen speziell für sie erstellt werden.

3 Wertschöpfungskette

3.1 Übersicht

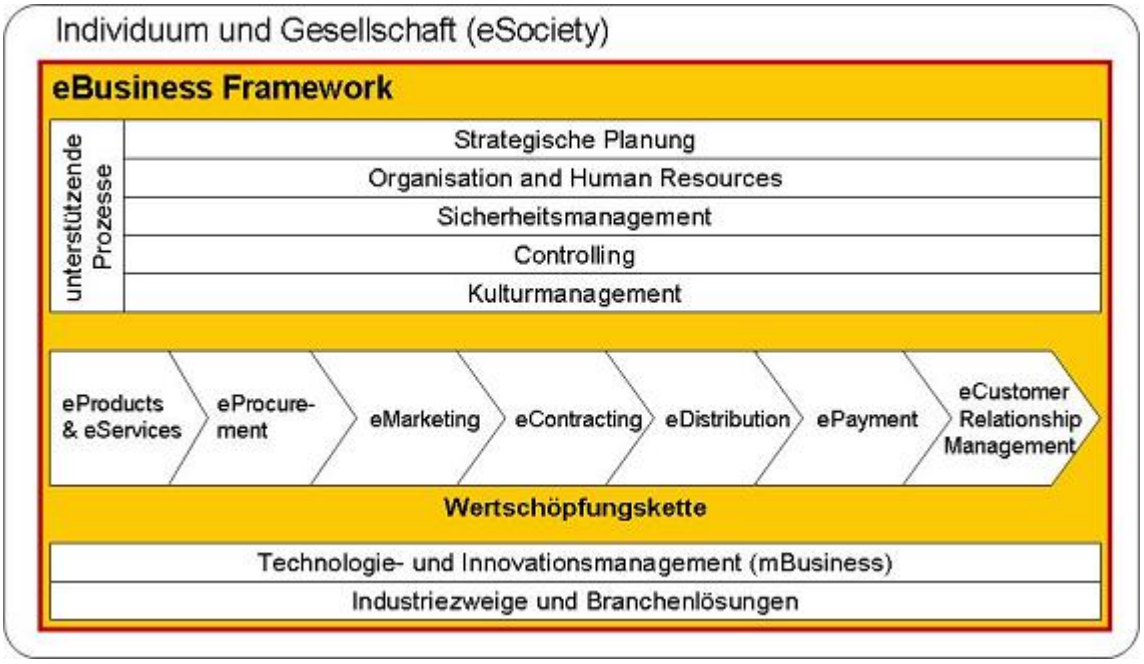


Abbildung 2: Wertschöpfungskette Quelle: [Meier/Stormer 2008]

3.2 Elemente der Wertschöpfungskette

3.2.1 eProducts & eServices

Für den Anbieter von Waren ist die Präsentation seiner Produkte ein zentraler Punkt. Die Darstellung, Beschreibung mit Attributen und Fotos und Gliederung ermöglicht es den potentiellen Kunden erst ein gewünschtes Produkt zu finden und bestenfalls auch zu bestellen. Für diesen Zweck muss der Produktkatalog ständig auf dem aktuellen Stand gehalten werden. Ein Aufwand, der bei grossen Systemn, die mehrere hunderttausend Artikel enthalten können, nicht zu unterschätzen ist. Bei komfortableren Systemen gibt es Möglichkeiten die Produkte hierarchisch in Warengruppen zu gliedern. Damit können einige Produktattribute eventuell schon auf der höheren Hierarchiestufe beschrieben werden und müssen nicht redundant erstellt und gewartet werden.
Je nach Produkt besteht für den Kunden auch die Option seine Artikel selber zu konfigurieren (zum Beispiel Dell: Konfiguration beziehungsweise Modifikation von Ausstattungsvorschlägen). Dann benötigen einzelne Komponenten noch weitere technische Angaben (Konfigurationsregeln), um technisch falsche oder sinnlose Kombinationen zu verhindern. Der Aufbau eines Konfigurationssystems ist natürlich aufwendiger und es hat auch Auswirkungen auf die Produktion und Distribution der Produkte. Sie können unter Umständen erst nach der Bestellung hergestellt werden, wenn es sich bei der Konfiguration nicht nur um eine sehr geringe Anzahl an Variationsmöglichkeiten handelt.

3.2.2 eProcurement

Unter *procurement* wird der Beschaffungsprozess in einem Unternehmen verstanden. Mit dem E-Zusatz versehen, handelt es sich dann um die durch spezielle Applikationen unterstützte elektronische Variante. Es werden generell 3 Varianten unterschieden:

- **Sell-Side:** Das System ist auf Verkäuferseite installiert. Der Aufwand mehrere Lieferantensysteme zu bedienen liegt beim Käufer.
- **Buy-Side:** Das System ist auf der Käuferseite installiert. Mit einem System können mehrere Lieferanten behandelt werden.
- und **Marktplatz:** Plattform mit mehreren Anbietern und Nachfragern. Der Betreiber ist in der Regel eine unabhängige Drittfirma .

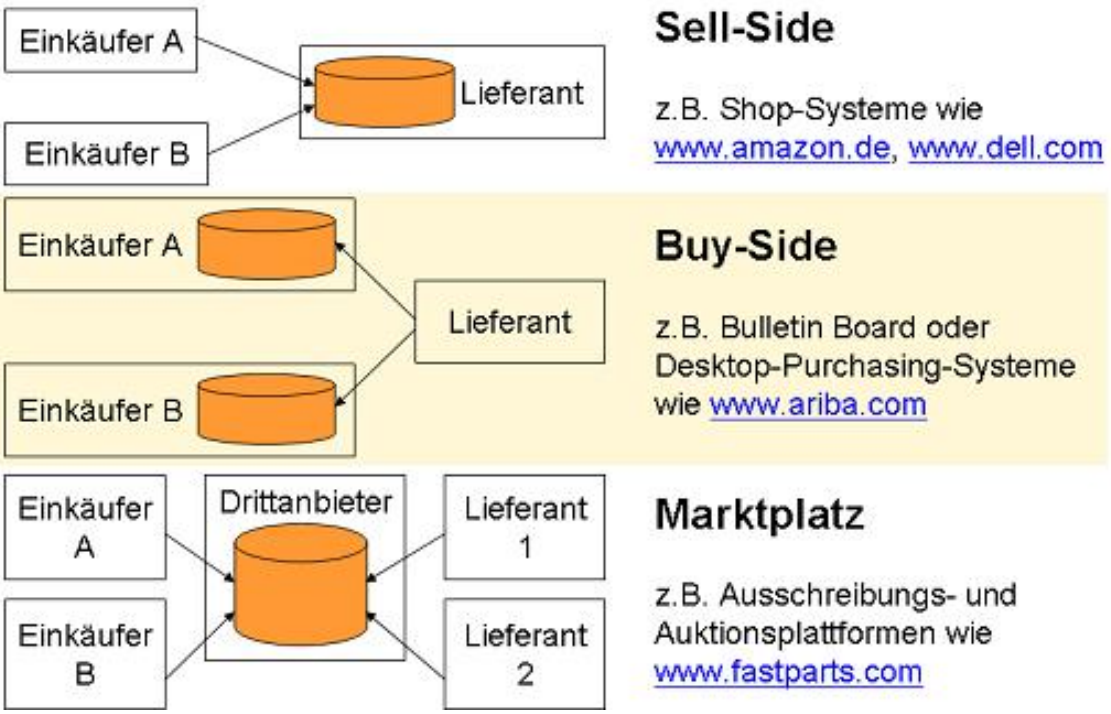


Abbildung 3: Marktmodelle eProcurement Quelle: [Meier/Stormer 2008]

3.2.3 eMarketing

Im Marketing gibt es die bekannte AIDA-Formel. Die Abkürzung steht für Attention, Interest, Desire und Action. So wie viele Erkenntnisse des klassischen Marketings ist auch diese Formel im eMarketing gültig. Natürlich gibt es zusätzlich jede Menge systemspezifische Merkmale. Die Einteilung der Kunden in Gruppen, die dann vom Marketing separat betrachtet und mit Massnahmen gesegnet werden können ist etwas anders als im klasssischen Marketing. Wir unterscheiden im eMarketing folgende Gruppen:

- Online Surfer

- Online Consumer
- Online Prosumer
  (Mischung aus Produzent und Consumer - trägt zur Wertschöpfungskette positiv bei und ist gleichzeitig auch Konsument)
- Online Buyer und
- Online Key Customer

Ein bei Webshops häufig vernachlässigtes Gebiet ist die Verkaufs-Psychologie. Wenn wir uns in Warenhäusern oder Supermärkten bewegen ist unsere gesamte Umgebung nach psychologischen Kenntnissen gestaltet. Das ist nicht nur die Akustik (also zumeist eine Hintergrundmusik), sondern kann auch olfaktorisch (Geruchsstoffe) und optisch über die Beleuchtung erreicht werden. Es ist zum Beispiel bekannt, dass die Gestaltung des Bodens (hart oder weich) einen starken Einfluss auf unsere Gehgeschwindigkeit hat und wird entsprechend eingesetzt. Eine unvollständige und etwas willkürliche Aufzählung finden Sie zum Beispiel unter http://www.orbit9.de/wissen/verkaufspsychologie.php. Mittlerweile gibt es aber auch im online-Bereich schon mehr Erfahrung, die auch in professionellen Systemen eingesetzt wird. Diese Erkenntnisse werden aber nicht sehr freizügig verbreitet. Wie sehr solche psychologischen Tricks dann trotzdem auf Personen wirken, die sie kennen ist dann noche eine weitere Diskussion. Das Wissen, dass zuviel Alkoholkonsum Kopfschmerzen verursacht verhindert den Konsum ja auch nicht vollständig.
Weitere wichtige Punkte im eMarketing sind cross-selling und up-selling, die noch im weiteren Verlauf des Skriptes behandelt werden.

3.2.4 eContracting

- **elektronischer Verhandlungsprozess**

  Softwaresysteme, die den elektronischen Verhandlungsprozess unterstützen, müssen die Dokumente und Unterlagen des gesamten Geschäftsprozesses verwalten und archivieren. Dazu gehören sämtliche Vereinbarungen und Vertragsabschlüsse, aber auch die digitalen Signaturen und das Controlling nach dem Vertragsabschluss.

- **digitale Signatur**

  Um rechtsgültige Vertragsabschlüsse im Internet zu erhalten, müssen sich die Geschäftspartner eindeutig identifizieren. Dies geschieht mittels digitaler Signaturen (siehe auch Kapitel 6.2.5 auf Seite 26). Dokumente, die mit gültigen Signaturen versehen sind, sind rechtswirksam.

3.2.5 eDistribution

Die Auswahl eines Distributionssystems sorgt für die Verteilung der Produkte, Waren oder Dienstleistungen an die Kunden. Die Definition des Distributionskanals legt fest, ob ein Produkt direkt oder indirekt abgesetzt wir. Beim indirekten

Absatz findet der Vertrieb nicht direkt zwischen dem Hersteller und dem Endkunden statt, sondern es gibt eine oder mehrere Zwischenhändlerstufen. Bei digitalen Produkten nennen sich die Zwischenhändler auch Infomediäre. Eine weitere strategische Aufgabe ist die Bestimmung der Distributionslogistik. Hierbei wird definiert, ob eine Ware gelagert wird oder in „just-in-time“ produziert wird, auf welchem Transportnetz und mit welchem Service die Verteilung stattfindet.

Dabei ist es von Bedeutung, ob wir mit materiellen Gütern oder mit digitalen Produkten handeln oder beide Formen im Sortiment führen.

- **online-Distribution**

  Bei einer reinen online-Distribution findet kein materieller Güterfluss statt. Musikvermarkter sind ein gutes Beispiel für diese Kategorie. Bei der online-Distribution können aber genau wie beim materiellen Güterfluss direkte und indirekte Absatzkanäle gewählt werden. Auch bei open-source Sotwareprodukten werden häufig indirekte Kanäle in Form von Spiegelservern verwendet, die von anderen Organisationen als dem Hersteller betrieben werden.

- **offline-Distribution**

  Für diesen Begriff gibt es mehrere Definitionen. So wird der Begriff offline-Distribution verwendet, wenn in Abgrenzung zur online-Distribution, ein materieller Güterfluss vorliegt. Dies kann durchaus auch bei digitalen Gütern sinnvoll sein, um Medien mit grosser Qualität zu übertragen oder zum Schutz der Urheberschaft. Ausserdem redet man von offline-Distribution, wenn Medien innerhalb eines Intranets zur Verfügung stehen, ohne dass ein Zugriff auf das Internet erfolgt. Das kann bei grossen Datenmengen wie Filmen oder Datenbanken sinnvoll sein.

- **hybride Distribution**

  Bei der hybriden Distribution werden on- und offline-Distribution kombiniert. Wenn die Produkte aus materiellen Gütern bestehen, können Zusätze wie Handbücher, Software, Firmware und so weiter separat elektronisch verteilt werden. Der Materiallieferung müssen dann keine CDs, DVDs oder sonstigen Datenträger mehr beiliegen. Damit ist auch eine höhere Aktualität bei Updates gewährleistet.

  Ein grosses Thema bei digitalen Produkten ist der Urheberrechtsschutz. Ohne besondere Vorkehrungen hat der Hersteller keine Kontrolle über die weitere Herstellung und Verteilung von Kopien. Bei Fotos können digitale Wasserzeichen eingesetzt werden, um die Urheberschaft zu kennzeichnen. Bei Softwareprodukten werden oft sogenannte Dongles[3] eingesetzt, die auf klassische Weise an den Kunden verschickt werden. Oft kann die Software nach dem Download für eine begrenzte Zeit ohne Einschränkungen verwendet werden und erst nach Ablauf einer Frist wird die Verwendung eines Product-Keys oder Dongles überprüft.

[3] Es werden auch die Begriffe Kopierschutzstecker, Hardlock oder Key verwendet. Die eingesetzte Software kontrolliert regelmässig die Existenz dieses Steckers.

3.2.6 ePayment

Die elektronische Abwicklung von Zahlungsvorgängen lässt folgendermassen klassifizieren:

- **Höhe des Betrages**

  Abweichend vom klassischen Handel, werden im Web Geschäfte abgehandelt, bei denen nur sehr kleine Beträge fällig werden - zum Beispiel für Webcontent. Das Problem bei Kleinstbeträgen ist es, den administrativen Aufwand so gering wie möglich zu halten damit die Kosten für die Buchungen nicht den Wert des Geldflusses übersteigen. Man unterscheidet bei der Höhe der Geldbeträge zwischen Nano-, Pico-, Micro- und Macropayment. Die genauen Betragsgrenzen sind nicht fix deklariert, sondern fliessend.

- **Zeitpunkt der Zahlung**

  Die Zahlung einer Dienstleistung oder eines Produktes kann vor, exakt bei der Übergabe oder nach der Übergabe erfolgen. Dies entspricht sinngemäss den drei Begriffen pre-paid (z. B. Geldkarten), pay now (z. B. Nachnahme) und pay later (z. B. Rechnung).

- **Anonymität**

  Genau wie im klassichen Handel, gibt es auch im elektronischen Geschäft die Möglichkeit anonyme oder nicht anonym zu bezahlen. Entscheidend ist dabei, ob der Käufer aufgrund des Zahlungsvorganges identifizierbar ist. Die klassische Bargeldbezahlung steht dabei für die anonyme Transaktion. Bei Kreditkarten, Rechnungen usw. kann der Käufer zumindest über mehrere Organisationsstufen eindeutig identifiziert werden.

- **Technik**

  Die eingesetzten Technologien lassen ebenfalls eine Differenzierung zu. Die Form der Abrechnung oder der Speicherung sind mögliche Kriterien.

4 Komponenten von Web-Applikationen

4.1 Usability

Der Begriff *Usability* ist umfassend und wäre auch ein Kandidat für eine Begriffsdefinition (aus der wir uns hier ja bekanntlich raushalten wollen). Es geht um gute Anwendbarkeit, klare Orientierung und Ausrichtung auf den Benutzer einer Applikation. Die Schwierigkeit bei Webapplikationen ist jedoch, dass der Benutzer meistens anonymer ist, als bei herkömmlichen Applikationen. Im Web wird eine grosse Anzahl potentieller Benutzer von vornherein durch die Sprache ausgegrenzt. Mittlerweile hat man zumindest erkannt, dass Anwender mit Behinderungen ebenfalls das Web benutzen und dass es die Effizienz steigert, wenn auch diese Benutzergruppen integriert werden können.

Eine gute Web-Applikation sollte ergonomisch gestaltet sein. Die folgenden Punkte geben einen Ansatz, worauf bei der Entwicklung besonders geachtet werden sollte.

Performance Die Antwortzeiten sollten – auch bei langsameren Internetverbindungen – angemessen sein. Vermeiden sie "schwere" Grafiken.

Verfügbarkeit Wie für *Stand-Alone*-Applikationen gilt auch auf dem Web: Das System darf nicht abstürzen und sollte sich korrekt verhalten, auch wenn viele Benutzer gleichzeitig zugreifen.

Plattformunabhängigkeit Die Applikation sollte auf allen (oder zumindest den meisten) gängigen *Webbrowsern* gut lesbar sein.[4] Beachten Sie, dass es mehr und mehr vollblinde Internetbenutzer gibt, die schlecht durch Grafiken navigieren können. Verwenden Sie bei Bildern die `alt`- und `title`-*Tags*.

Lesbarkeit Die Schrift sollte gut lesbar sein: Schriftgröße und Kontrast auf diversen Systemen prüfen. Hintergrundgrafiken können die Lesbarkeit extrem vermindern.

Seitenlänge Auf Internetseiten sollte möglichst wenig *gescrollt*[5] werden. Schreiben Sie nicht: `Siehe weiter unten im Text...` sondern verwenden Sie lieber einen sogenannten *Hyperlink* auf eine neue Seite. Auf der Hauptseite (*home*) sollte **nie** gescrollt werden müssen. Vorteile von kurzen Seiten: Bessere Navigierbarkeit, bessere Strukturierung möglich, schnellere *download*-Zeiten, bessere Wartbarkeit.

Orientierung Der Benutzer sollte jederzeit sehen, wo er sich gerade in der Informationshierarchie befindet. Verwenden Sie immer das `<title>`-*Tag* im Seitenkopf, damit die User die Seite in den Favoriten (bzw. Bookmarks) einfach wieder finden können. Auf jeder Seite eine Navigationshierarchie[6] anzugeben ist auch sinnvoll:

`MyShop > Zahlungsmodalitäten > Rechnung > Howto`

[4] Internet Explorer, Netscape, Mozilla, Opera, Konqueror, ...
[5] rollen mittels Rollbalken
[6] Breadcrumb-trails:
(`http://psychology.wichita.edu/surl/usabilitynews/52/breadcrumb.htm`)

Konsistenz Alle Seiten einer Web-Applikation sollten sich immer gleichartig verhalten. Die Grafiken und Anordnungen der Steuerelemente sollten einheitlich sein. Verwenden Sie Fluchtlinien, um das Auge zu beruhigen. Beachten Sie, dass auch allfällige Bannerwerbung zum Layout passt.

Verwenden Sie für Abstände, Rahmen, Farben, Schrift und so weiter wenn immer möglich CSS[7]-Vorlagen. Die Applikation wirkt einheitlicher, ruhiger und professioneller. Zudem können Sie diese viel einfacher **anpassen**.

Didaktik Das System sollte leicht erlernbar sein – wenn möglich ohne Einführung oder Hilfe-Seiten. Alle Informationen und *Links*, die benötigt werden, um den nächsten Schritt (Warenkorb ansehen, Artikel zukaufen, Zahlungsbedingung aushandeln, ...) auszuwählen, sollten permanent ersichtlich sein.

Natürlich ist für *B2B* (bzw. *B2E*) Applikationen die Usability nicht im Vordergrund. Hier sollte schnell gearbeitet werden können. Oft wird eine gewisse Einarbeitung in Kauf genommen, um danach eine noch höhere Performance zu erreichen.

4.2 Anpassungen - Change Management

Vermutlich nicht ganz unbeabsichtigt, kommt im Titel des Moduls "E-Business-Applikation anpassen" das Wort **anpassen** vor. Ab wann genau etwas eine Anpassung ist oder eine Neuentwicklung ist leider wieder eine Definitionsfrage. Durch die Verwendung von Frameworks und Generatoren sind auf Neuentwicklungen auf Codebasis eigentlich nur Anpassungen. Wenn es sich aber um eine Applikation handelt, die bereits produktiv im Einsatz ist, dann unterscheidet sich das Vorgehen von einer Neuentwicklung. Vor allem die Überführung aus der Testumgebung in die produktive Umgebung ist in diesem Fall heikel, da der bestehende Applikationseinsatz ungehindert und fehlerfrei weiterlaufen soll. Laut ITIL [8] soll dies kontrolliert, effizient und unter Minimierung der Risiken geschehen.

Das Change-Management ist dafür verantwortlich standardisierte Prozesse zu definieren, die bei Änderungen der IT-Infrastruktur angewendet werden. Gerade bei *„running-systems"* kann der Schaden erheblich sein, wenn nach Änderungen Störungen auftreten oder das System im Extremfall ganz ausfällt.

Die Anpassungen können darin bestehen, dass vorgegebene Parameter verändert werden - verändern der Einstellungen auch Customizing genannt oder dass Codeanpassungen durch Anfügen, Entfernen oder Modifizieren bestehender Funktionenberücksichtigen gemacht werden.

Ressourcen:

- `Wikipedia zum Thema Change Management`
- `http://www.change-management.com`

In der Praxis werden häufig Versionierungssysteme wie CVS [9] oder Subversion eingesetzt. Da sich dieser Kurs nicht mit dem Installieren einer Versionsverwaltung aufhalten will, wählen wir das sogenannte *"Poor-mans-CVS"*: Wir speichern mit jeder Version alle Dateien in einem neuen Verzeichnis ab.

[7] Cascading Style Sheets
[8] IT Infrastructure Library
[9] Concurrent Versions System

4.3 Session

Ein generelles Phänomen von Multi-User Applikationen (und somit im speziellen auch von Web-Applikationen) ist die Tatsache, dass jeder Benutzer seine eigenen Variablen besitzen muss.

Sessions (Sitzungen) implementieren auf verbindungslosen Protokollen (hier `http`) eine Beziehung vom Client zu einem zugehörigen Prozess auf dem Server. Es wird dem Client eine ID[10] zugewiesen. Jeder nun folgende Aufruf des Clients übergibt dem Server diese ID. Somit kann der Server seine Prozesse und Daten eindeutig den Clients zuweisen. Eine andere Möglichkeit wäre es, alle Nutzdaten auf dem Client abzulegen.

Mögliche Implementationen über `http`:

Cookies Speichern von Variablenwerten durch den Browser auf dem Client. Ein "Cookie" enthält den Namen, den Inhalt (Content), die *Domain*, den Pfad, ein Verfallsdatum, eine *Policy* und den Sicherheitszustand (*secure*).

URL-Rewriting Beim URL[11]-Rewriting wird jedem Hyperlink eine Identifikationsnummer mitgegeben
(z. B. `<a href="http://www.xyz.com?SessionID=Sz45Km232Kxw22">`).

Hidden-Fields Variablen können auch in versteckten Feldern der Formulare (im `<form>`-Element) mitgegeben werden.

Überlegen Sie sich die Sicherheitsrisiken oder allfällige Datenschutzproblematiken der drei genannten Verfahren.

In `PHP` wird eine Session einfach mit `session_start();` generiert. Die Applikationsentwickler müssen sich nicht mehr um das darunterliegende Verfahren kümmern.

4.4 Personalisierung

Individuelle Ansprache von bekannten Kunden: *Guten Tag Herr Gressly, gerade eben frisch eingetroffen das neueste PHP-MySQL Buch!* Auch könnte es möglich sein, dass ein Benutzer sein *Layout* (Aussehen) und das Verhalten der Applikation bis zu einem gewissen Grad selbst bestimmen kann.

4.5 Bannerwerbung

Bannerwerbungen sind für die Kunden lästig. Es ist jedoch eine einfache Möglichkeit, Werbefläche für bares Geld zu verkaufen. Die Werbefläche kann auch *per Click* oder *per Show* verkauft werden.

[10] Identifikationsnummer oder -string
[11] Uniform Resource Locator

4.6 Web-Shop Konzepte

Im Folgenden soll speziell auf Konzepte des Web-Shops eingegangen werden. Selbstverständlich finden wir einige dieser Konzepte in ähnlicher Form in anderen Applikationen wieder.

4.6.1 Passantenfunktion

Passantenfunktionen werden alle Funktionen genannt, die die Besucher eines Webshops ohne Anmeldung ausführen können. Aus der Sicht des Anbieters ist es ein zweischneidiges Schwert. Auf der einen Seite ist die Identifizierung eines Shopbesuchers für das eMarketing sehr wichtig. Nur durch eine genaue Zuordnung der online-Aktivitäten zu Benutzern können detaillierte Kundenprofile erstellt werden. Auf der anderen Seite möchten viele Shopbesucher genau das verhindern und wünschen keine umfangreiche Analyse ihres Verhaltens in deren Ergebnisse sie ja noch nicht einmal Einblick erhalten. Um diese potentiellen Käufer trotzdem nicht zu verlieren, ist es in vielen Shops möglich einen direkten Kauf zu tätigen ohne ein Kundenkonto zu eröffnen. Für den Kunden entfallen dann diverse Vereinfachungen wie Einmalerfassung seiner Adresse, Erstellung von persönlichen Profilen (My-Account), Übersicht über alle Bestellungen und so weiter. Wer aber bei einem unbekannten Lieferanten nur voraussichtlich einmal etwas kaufen möchte, hofft so eventuell nicht in die permanente Kundenstammdatenbank aufgenommen zu werden. Wenn das Produkt per eDistribution bezogen wird und die Zahlung mit einer anonymen Methode erfolgt, kann er mit dieser Annahme durchaus richtig liegen. Falls aber doch eine Identifizierung erfolgt, ist es für den Anbieter zwar aufwendiger ein Profil zu erstellen, aber dennoch möglich.
Für Passantenfunktionen erfolgt also kein Login und die Daten über Zahlungsart und Lieferadresse muss der Kunde - wenn überhaupt - erst ganz am Schluss eingeben, wenn er wirklich etwas kaufen will.

4.6.2 Preisfindung

Preise werden im Warenkorb anhand der Artikel zusammengestellt. Jedoch können hier je nach Kunde und aktuellen Rabatten andere Resultate entstehen.

Beispiel 1 *Kunden, die im letzten Monat für mehr als CHF 200.- eingekauft haben, erhalten einen Sonderrabatt.*

Kunden, die mehr als 5 Artikel im Warenkorb haben, erhalten 10% Rabatt.

...

4.6.3 Auftragsbestätigung

Bevor eine Bestellung wirklich ausgeführt wird, hat der Kunde die Möglichkeit, alle Artikel, Preise, Rechnungs- und Lieferanschriften, Rabatte und so weiter anzuschauen. Der Kunde sieht alle Produkte, die Lieferadresse und die Zahlungsart noch einmal, bevor er zuallerletzt auf den Schalter `Auftrag versenden` klickt.

4.6.4 Cross- und Up-Selling

Mit diesen beiden Begriffen (Cross-Selling und Up-Selling) sind Angaben gemeint, die den Kunden darauf hinweisen, dass es noch verwandte Artikel (höherwertige, ergänzende) zum aktuellen Produkt gibt. Es können auch gleich Rabattpakete mit mehreren Artikeln angeboten werden.

4.6.5 Data-Mining

In großen Warenhäusern werden Statistiken erzeugt, die Zusammenhänge im Kaufverhalten aufdecken, um besseres Cross- bzw. Upselling zu betreiben. Ebenso kann damit personalisiert und Aktionen können sinnvoll geplant werden.

4.6.6 Warenkorb

Der Warenkorb ist eine zentrale Funktion des Webshops. Wichtig ist zu wissen, wie die Inhalte im Warenkorb gespeichert sind. Die Variante, die Inhalte *clientseitig* zu speichern, birgt Risiken. Auf der Serverseite gibt es zwei Varianten: a) persistent: Der Inhalt des Warenkorbes wird in einer Datenbank gespeichert und b) transient: Der Warenkorb lebt in der "Session" als temporäre Variable. Hier muss man sich überlegen, ob die Waren auch noch nach längerer Zeit im Korb liegen sollten. Oder macht es Sinn, die Session-Variablen nach einer halben Stunde – mit allen Waren im Korb – zu löschen. Die Anwenderin muss sich dann wieder neu anmelden.

4.6.7 Produkte Auswahl

Um Produkte eines Web-Shops zu finden, gibt es diverse Strategien. Wichtig ist, dass der Kunde rasch auf das gesuchte Produkt stößt. Das kann bei kleinen Anbietern eine einfache Tabelle mit allen im Lager befindlichen Artikeln sein. Eine **Suche** nach Artikeln kann diverse Stichworte berücksichtigen oder aber nach allen im Text vorkommenden Wörtern suchen (Index). Häufig werden auch Produktehierarchien angeboten. So kann sich ein Käufer wie im Supermarkt von Stockwerk zu Stockwerk und anschließend von Regal zu Regal bewegen, bis er beim gewünschten Produkt ankommt.

5 Sicherheit

5.1 Angriffe

Neben den normalen Problemen des Datenverlusts und der Fehlmanipulation gibt es eine Fülle von *bösartigen* Angriffen auf eine Web-Applikation. Überlegen Sie zu den folgenden Angriffen je eine Gegenmaßnahme.

5.1.1 Spionage

Um an vertrauliche oder wertvolle Firmendaten zu gelangen gibt es eine Menge Tricks. Eine Variante ist das (illegale) herunterladen ganzer Datenbestände von Web-Applikationen. Wie können wir uns dagegen wehren?

Beispiel 2 (GET-IDs) *Eine einfache Webapplikation findet die Artikel oder Produkte mit einem GET-Parameter:* `http://www.xyz.com?art_id=35`*. Somit kann ein Angreifer alle Artikelnummern*[12] *ausprobieren und die ganze Datenbank in kurzer Zeit ausspionieren.*

5.1.2 Passwort Cracker / Passwort Guesser

Wer an einem System genügend *Logins* durchführen darf, kann mit einem einfachen, jedoch zeitaufwändigen Verfahren Passwörter herausfinden. Sogenannte "Cracker"-Angriffe probieren *brute-force*[13] alle Möglichkeiten durch. Password--Guesser hingegen versuchen aufgrund von Daten des Benutzers (Geburtsdatum, Namen von Verwandten, Beruf, ...) an mögliche und sinnvolle Passwörter heranzukommen.

Meistens geschehen Passwort-Angriffe aber nicht von extern (also von außerhalb der Firma). Wer im Besitz einer Passwortliste ist, kann autorisierten Zugriff auf verschiedene Konten erhalten.[14]

Unsichere SQL-Anfragen Eine spannende Art, sich Zugang zu einem Webserver ohne Rechte zu verschaffen, funktioniert mittels *unsicheren* SQL-Abfragen. Häufig werden Anfragen an die Datenbank wie folgt gestellt (hier ein PHP Beispiel):

```
<?php
  $sql="SELECT * FROM user WHERE name='$uname'";
  $result=mysql_query($sql);
?>
```

Ein schlauer Benutzer könnte nun bei der Anfrage nach seinem Benutzernamen ins Feld einfach folgenden Eintrag tätigen.

[12] `art_id=1, 2, ...`

[13] "Mit aller Kraft", stur, durch simples Probieren aller Varianten

[14] Das funktioniert oft auch für verschlüsselte Passwortlisten, da diese mit sogenannten öffentlich bekannten *Hash*-Funktionen arbeiten.

Benutzer: `blah'; UPDATE user SET password='simple`

Falls dieser Benutzername eins-zu-eins in die Variable `$uname` eingesetzt wird, so lautet die SQL-Anfrage nun wie folgt:

```
$sql="SELECT * FROM user WHERE name='blah';
      UPDATE user SET password='simple'";
```

Der Benutzer ist zwar damit noch nicht *eingeloggt*, aber das Passwort wurde bei allen Benutzern nach `'simple'` abgeändert. Somit kann ein späteres *Einloggen* nicht wirklich schwierig sein.

Abhilfe Abhilfe verschafft man sich, indem vor alle Apostrophe, die vom Benutzer eingegeben wurden, ein `Back-Slash \` vorangestellt wird[15]. Somit kann ein SQL-Statement nicht mehr mutwillig beendet werden.

5.1.3 Horcher und "The-Man-in-the-Middle"

Ein Abhorcher (Horcher) schaut sich den ganzen Datentransfer zwischen zwei Sockets an und versucht so, Informationen über die Schwachstellen des Systems zu erhalten. Später kann er mit diesem Wissen das System direkt angreifen.

Ein *Man-in-the-Middle* dagegen fängt den gesamten Verkehr zwischen zwei Systemen ab, modifiziert den Inhalt und sendet die Änderungen ans Gegenüber. Somit ist es einem *Man-in-the-Middle* z. B. möglich, einem System vorzugaukeln, er sei ein legaler Kunde. Ein solches Einschleusen funktioniert nur bei "langsamen" Transaktionen (z. B. E-Mail), wo die Endpunkte nichts von der Verzögerung (die durch die Veränderung am Inhalt entsteht) mitbekommen.

5.1.4 E-Shop Lifting

Falls es einem Angreifer möglich ist, die Preise auf einer Webseite zu verändern und so Angebote billiger zu erschleichen, so sprechen wir von **E-Shop Lifting** oder auch von "virtuellem Ladendiebstahl" [CT Nr. 26 2002]. Das funktioniert z. B. dann, wenn der "Shop" die Preise in *Hidden-Fields* auf dem Client ablegt.

5.1.5 Session Hijacking

Ein Angreifer *übernimmt* eine bestehende Sitzung. Dieses Vorgehen wurde bei TCP[16] eingehend untersucht. Natürlich ist dies bei einfachen SessionIDs keine Hexerei.

5.1.6 Viren, Würmer und anderes Getier

Viren, Würmer und Enten (Hoax) gefährden in erster Linie die Endanwender und nicht die Web-Applikation. Es gibt jedoch immer wieder Fälle, wo auch die Webserver mehr oder weniger gezielt attackiert werden.

[15] PHP kennt hier die Methode `addslashes()`
[16] Transfer Control Protocol

5.1.7 DoS Attacken, Trojaner und Hintertüren

Webserver werden eher Ziel einer **Denial of Service**-Attacke (DoS) als der gemeine Heimanwender. Dabei machen mehrere PCs gleichzeitig simple Anfragen an einen Webserver. Dieser Server wird dann durch die Fülle von Anfragen lahmgelegt.

Um DoS-Attacken vorzubereiten, werden häufig sogenannte **trojanische Pferde**[17] eingesetzt. Diese setzen, ohne das Wissen des PC-Besitzers, zu einem bestimmten Zeitpunkt Anfragen auf das Opfer der DoS-Attacke ab.

Trojanische Pferde können aber auch eingesetzt werden, um **Hintertüren** (sog. *Backdoors*) zu öffnen. Mit offenen Hintertüren ist es einem entfernten Angreifer möglich, alle Information über das System zu erhalten und dieses auch nach seinen Wünschen zu modifizieren.

5.2 Autorisierung/Authentifizierung

Um auf einem entfernten System arbeiten zu können, braucht es Zugriff (Authentifizierung) und Berechtigungen (Autorisierung).

Die Authentifizierung geschieht im Normalfall mit Passwörtern. Es wird unterschieden zwischen schwacher (allein mittels Passwörtern) und starker Authentifizierung. Letztere benötigt *something to know* **und** *something to have* (Passwort **und** Streichliste). Die Autorisierung (Bevollmächtigung mit Privilegien) geschieht nach der Authentifizierung.

Definition 1 (Authentifizierung) ***Authentifizieren*** *heißt: "Die Echtheit von etwas bezeigen, beglaubigen." [Wahrig] In der Informatik wird ein Benutzer oder ein System (Software, Client, ...) authentifiziert. Der Server will wissen,* ***wer*** *den Dienst in Anspruch nimmt. Dieses Wissen über das Gegenüber erlaubt z. B. eine Autorisierung oder eine finanzielle Abrechnung. Zur starken Authentifizierung kann mittels Streichlisten oder Secure-IDs vorgegangen werden.*

Definition 2 (Autorisierung) *Das WAHRIG Fremdwörterlexikon umschreibt* ***Autorisierung*** *mit "Bevollmächtigung". In verteilten Systemen ist es wichtig, dass nur ermächtigte Personen Privilegien auf bestimmten Daten erhalten: hinzufügen, suchen, ansehen, löschen, verändern, vergeben weiterer Rechte, ... Hier geht es darum,* ***was*** *eine Person oder ein System tun darf.*

5.3 Kryptographie

Kryptographie bezeichnet die Verschlüsselung von Daten, um diese vor fremder Einsicht zu schützen. Ebenso soll auch die Echtheit des Absenders überprüft werden können, und seine Rechte müssen definiert werden. Verfahren zur Kryptographie werden im Kapitel Kryptographie (siehe auch Kapitel 6 auf Seite 24) genauer behandelt.

[17] Griechische Mythologie (Ilias): Die Griechen eroberten die Stadt Troja mit Hilfe des hölzernen Trojanischen Pferdes, in dessen hohlem Bauch sich die tapfersten Helden verbargen und so von den ahnungslosen Trojanern in die Stadt geführt wurden.

6 Kryptographie

6.1 Überblick

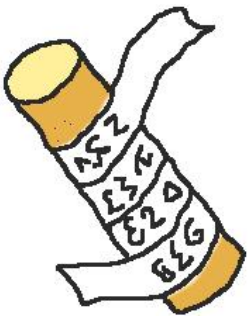
- In diesem Dokument wird zunächst mit historischen Verfahren und auch mit dem XOR-Verfahren (siehe auch Kapitel 6.2.3 auf Seite 25) begonnen. Danach werden einige mathematische Grundlagen gegeben. Diese Grundlagen werden dann im Verfahren von Diffie/Hellman und RSA (siehe auch Kapitel 6.5 auf Seite 33) angewendet. Im letzten Abschnitt (siehe auch Kapitel 6.7 auf Seite 36) wird noch erklärt, wie die mathematischen Hilfsprogramme (JAVA) zu bedienen sind. Denn: wer rechnet schon gerne `;-)`

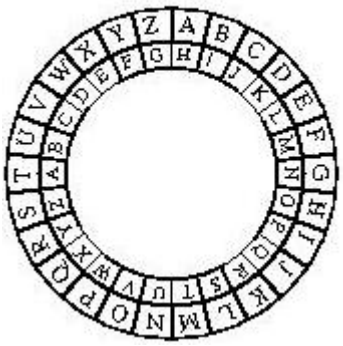
- Wer sich weder für die Geschichte der Kryptographie noch für die mathematischen Hintergründe interessiert, kann gleich zur Anwendung der Verfahren blättern: Diffie/Hellman (siehe auch Kapitel 6.4 auf Seite 31) / RSA (siehe auch Kapitel 6.5 auf Seite 33).

- Mathematische Verschlüsselungsverfahren, die auf großen Primzahlen basieren, haben die Stärke, dass genau berechnet werden kann, wie groß der durchschnittliche (zeitliche, rechnerische) Aufwand sein wird, um eine Botschaft unrechtmäßig zu "entschlüsseln".

6.2 Geschichte

Beginnen wir mit dem Verfahren der Spartaner (oder Babylonier?). Sie verwendeten bereits im 5. Jh. v. Chr. zur Verschlüsselung einen runden Stab und einen Papierstreifen (Linke Abbildung).





Nur wer einen Stab der selben Dicke besaß, war in der Lage, derart verschlüsselte Botschaften zu entziffern.

Überlegen Sie sich, mit welchem Aufwand dieses Verfahren "geknackt" werden kann.

6.2.1 Cäsar

Der Algorithmus von Cäsar ist nicht viel sicherer (Rechte Abbildung). Das Verfahren verschiebt die Buchstaben im Alphabet um eine vorgegebene Anzahl. Es kommen 26 Buchstaben im Alphabet vor, somit gibt es lediglich 26 mögliche Schlüssel. Bedenken Sie aber, dass um 50 v. Chr. noch fast niemand lesen oder schreiben konnte. Somit war das Verfahren für die damalige Zeit sicher genug.

Beispiel 3 (Cäsar) *Der Schlüssel sei 7. Das heißt, jeder Buchstabe wird im Alphabet um 7 Zeichen nach hinten verschoben.*
"Hallo" $\Rightarrow$ *"Ohssv".*

6.2.2 Monoalphabetische Substitutionen

A	B	C	D	E	F	G	...
x	π	σ	r	ϱ	!	ϑ	...

Häufig wird anstelle einer Translation des Alphabetes eine beliebige Permutation (Vertauschung) verwendet. Ob man wieder Buchstaben oder irgendwelche kryptisch anmutenden Zeichen verwendet, spielt hier keine Rolle.

Das Verfahren kann jedoch auch sehr rasch *geknackt* werden. Wie wohl?

Bemerkung: Das Verfahren von Cäsar ist ein Spezialfall einer monoalphabetischen Substitution.

6.2.3 Das XOR-Verfahren

Wie funktioniert das XOR-Verfahren? Wir gehen davon aus, dass eine Datei verschlüsselt werden sollte. Diese Datei können wir einfach in eine Byte- bzw. eine Bitfolge verwandeln (wir können z. B. Unicode verwenden). Wenn nun der Sender wie auch der Empfänger vorab eine zufällige Bitfolge (Schlüssel) ausgetauscht hatten, so können Sie die Botschaft bitweise mit der exklusiven Oder-Operation (XOR) verknüpfen.

Das XOR-Verfahren ist absolut sicher, wenn wir davon ausgehen, dass der Schlüssel genug streut (die Null- und Einsbits sind rein zufällig gewählt). Jetzt kann das Verfahren nicht mehr geknackt werden. Um an die Information zu kommen, müsste schon der Schlüssel "geraubt" werden.[18]

Wichtig ist auch zu wissen, dass ein XOR-Schlüssel nur einmal eingesetzt werden sollte.

Das XOR-Verfahren kann z. B. auch mit einem Pseudozufallszahlen-Algorithmus gestartet werden. Hierbei ist der Schlüssel eine sogenannte "Random-Seed"-Zahl. Wenn zwei gleich gebaute Zufallszahlengeneratoren mit demselben Startwert beginnen, so liefern sie auch dieselbe Zahlenfolge. Das hat den Vorteil, dass nur eine kleine Information ausgetauscht werden muss. Das Verfahren verliert dabei aber an Sicherheit!

[18] Ein analoges Verfahren zum XOR-Verfahren ist der *One Time Pad* von AT&T (1917).

6.2.4 Public Key

James H. Ellis hat 1970 ein Verfahren entwickelt, bei dem Schlüssel, oder zumindest Teile davon, öffentlich übermittelt werden können. Auch wer diese Schlüsselteile kennt, kann Botschaften noch nicht genügend rasch knacken. Die Idee ist brillant, doch – werden Sie sich fragen – geht das überhaupt?

In der Regel wird bei einem solchen Public-Key[19]-Verfahren eine Rechenoperation eingesetzt, die nicht einfach umzukehren ist. Denken Sie z. B. an das Wurzelziehen aus der Grundschule. Das Multiplizieren zweier Zahlen geht rasch und einfach. Jedoch die Quadratwurzel einer Zahl zu bestimmen, braucht schon wesentlich größeren Aufwand. Auf einer ähnlichen Grundidee basieren moderne Krypto-Verfahren.

Erst mittels solcher "Einwegfunktionen"[20] wird es möglich, dass zwei Parteien, die vorher noch nie miteinander in Kontakt getreten sind, geheime Botschaften austauschen! Bisher mussten (wie beim XOR-Verfahren) die Parteien vorher einen Schlüssel über einen geheimen Kanal verschicken!

6.2.5 Digitale Signatur

Die digitale Signatur entspricht einer Unterschrift oder einem Siegel. Nur wer den Siegelring (hier den Private-Key) besitzt, kann die Signatur anfertigen.

Im Gegensatz zur Geheimhaltung bleibt beim digitalen signieren die Botschaft unverschlüsselt. Es wird lediglich ein Hash-Code (siehe auch Kapitel 6.3.5 auf Seite 29) der Botschaft **verschlüsselt**; und zwar diesmal mit dem **Private-Key**.

Alle sollen die Herkunft der Botschaft überprüfen können. Hierzu wird mit dem Public-Key der verschlüsselte Hash-Code dechiffriert und mit dem Hash-Code der unverschlüsselten Botschaft verglichen. Da der Public-Key öffentlich zugänglich sein soll, ist es für jede Person möglich, die Unterschrift auf Echtheit zu prüfen; vorausgesetzt natürlich, dass bereits dem Public-Key vertraut werden kann `;-)`

6.3 Mathematische Grundlagen zu Krypto-Verfahren

- Dieses Kapitel beleuchtet die mathematischen Hintergründe, die für die Anwendung der Verfahren RSA, ElGamal und Diffie/Hellman notwendig sind.
- Diese Einführung erhebt keinen Anspruch auf Vollständigkeit. Insbesondere werden wichtige Beweise weggelassen.
- Es geht in den nachfolgenden Kapiteln lediglich darum, dass die beiden Verfahren RSA und Diffie/Hellman in groben Zügen verstanden und angewendet werden können.
- Trotz meinem pragmatischen Ansatz werden einige Grundlagen der Zahlentheorie eingeführt:

[19] Public-Key = öffentlicher Schlüssel

[20] Im Gegensatz zu Hash-Funktionen (siehe auch Kapitel 6.3.5 auf Seite 29) sind diese Funktionen umkehrbar. Jedoch ist der Aufwand, die Funktion umzukehren enorm viel höher, als die Funktion zu berechnen.

6.3.1 ggT

Der "ggT" von natürlichen Zahlen, ist der **größte gemeinsame Teiler**. (engl. GCD = greatest common divisor). Def.: Der größte gemeinsame Teiler von zwei Zahlen ist die größte ganze Zahl, die beide Zahlen ohne Rest teilt.

Beispiel 4 (ggT) *größter gemeinsamer Teiler:*

```
ggT (48, 32) = 16.
ggT (10, 11) = 1
ggT (50, 70) = 10
ggT (35, 63) = 7
```

Bemerkung 1 *Zwei Zahlen a und b sind genau dann teilerfremd, wenn* $ggT(a, b) = 1$ *ist.*

Um den ggT von zwei großen Zahlen zu berechnen, verwenden Sie das Programm GCD:

```
>java GCD i j.
```

6.3.2 Primzahlen

Eine **Primzahl** ist eine Zahl, die neben sich selbst nur die 1 als Teiler hat. Mit anderen Worten: Eine Primzahl hat genau zwei Teiler. Die kleinste Primzahl ist 2.

Beispiel 5 (Primzahlen) *Hier die ersten zehn Primzahlen: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...*

Bemerkung 2 *Ist n eine beliebige (positive, ganze) Zahl, und ist p eine Primzahl, so gilt:*

$$ggT(n, p) = \begin{cases} p, \text{ wenn } p \text{ Teiler von } n \text{ ist.} \\ 1, \text{ wenn } p \text{ kein Teiler von } n \text{ ist.} \end{cases}$$

Für unsere Aufgaben müssen wir Primzahlen finden - je größer umso besser. Beispiele von Primzahlen finden wir z. B. auf dem Internet unter:

```
http://www.geocities.com/primes_r_us/small/index.html
```

Natürlich werden für sehr sichere Verschlüsselungen weitaus größere Primzahlen (ab 300 Stellen) verwendet. Abgesehen davon, dass die Primzahlverfahren sehr sicher sind, weisen sie noch eine zusätzliche Stärke auf: Es ist berechenbar, mit welchem durchschnittlichen Zeit- bzw. Rechenaufwand die Verfahren geknackt werden können.

6.3.3 Modulo p

Mit *Modulo* (mod) bezeichnen wir das Berechnen von Divisionsresten.

$m \bmod p :=$ Rest, der entsteht, wenn wir m durch p teilen.

Beispiele:

1. $17 \bmod 7 = 3$ (denn $17 = 2 * 7 + 3$)
2. $37 \bmod 5 = 2$ (denn $37 = 6 * 5 + 2$)

Bemerkung 3 (Distributivität von Modulo-Berechnungen) *Ist p eine Primzahl, so gelten folgende Regeln:*

$$(a+b) \bmod p = [(a \bmod p) + (b \bmod p)] \bmod p \quad (1)$$
$$(a*b) \bmod p = [(a \bmod p) * (b \bmod p)] \bmod p \quad (2)$$
$$(a)^n \bmod p = [(a \bmod p)^n] \bmod p \quad (3)$$

Beispiel 6 (Modulo Beispiele)

$$\begin{aligned}
(5+2) \bmod 3 &= ((5 \bmod 3) + 2) \bmod 3 = (2+2) \bmod 3 = 4 \bmod 3 = 1 \\
(8*15) \bmod 7 &= (1*1) \bmod 7 = 1 \bmod 7 = 1 \\
(14*35) \bmod 3 &= (2*2) \bmod 3 = 4 \bmod 3 = 1 \\
46^{10} \bmod 11 &= 2^{10} \bmod 11 = 4^5 \bmod 11 = (4*4^4) \bmod 11 = (4*16^2) \bmod 11 \\
&= (4*5^2) \bmod 11 = (4*25) \bmod 11 = (4*3) \bmod 11 = 1
\end{aligned}$$

Bemerkung 4 (RSA) *Für das RSA - Verfahren (siehe auch Kapitel 6.5 auf Seite 33) und das Verfahren von Diffie/Hellman(siehe auch Kapitel 6.4 auf Seite 31) brauchen wir* $a^b \bmod c$ *zu berechnen. Berechnen Sie zum Beispiel* $23^{17} \bmod 7$ *mit dem* JAVA *Hilfsprogramm* `AhBmC`*:*

```
>java AhBmC 23 17 7
```

6.3.4 Inverses Modulo p

Für das RSA-Verfahren benötigen wir noch die folgende Rechenoperation. Wenn Sie sich nur für Diffe-Hellman oder das Verfahren von ElGamal interessieren, so können Sie dieses Kapitel überblättern.

Wenn wir Modulo p rechnen und p eine Primzahl ist, so gibt es für jede Zahl m ein sogenanntes multiplikatives inverses Modulo p. Das heißt: für jedes m gibt es ein n, sodass $m * n \pmod p = 1$ ist.

$p = 7 \quad m = 3 \rightarrow n = 5$ (denn $3*5 \bmod 7 = 1$)
$p = 7 \quad m = 6 \rightarrow n = 6$ (denn $6*6 \bmod 7 = 1$)
$p = 11 \quad m = 5 \rightarrow n = x$ (berechnen Sie x selbständig)

Hier ein simples Vorgehen, um das Inverse (mod p) zu finden:

Nehmen wir z. B. $p = 13$ und $m = 5$

$$
\begin{array}{rcrcrcr}
5 & * & 2 \bmod 13 & = & 10 \bmod 13 & = & 10 \\
5 & * & 3 \bmod 13 & = & 15 \bmod 13 & = & 2 \\
5 & * & 4 \bmod 13 & = & 20 \bmod 13 & = & 7 \\
5 & * & 5 \bmod 13 & = & 25 \bmod 13 & = & 12 \\
5 & * & 6 \bmod 13 & = & 30 \bmod 13 & = & 4 \\
5 & * & 7 \bmod 13 & = & 35 \bmod 13 & = & 9 \\
5 & * & \mathbf{8} \bmod 13 & = & 40 \bmod 13 & = & \mathbf{1}
\end{array}
$$

Daraus ergibt sich 8 als das Inverse (mod 13) zu 5, denn $5 * 8 \bmod 13 = 1$.

Das JAVA Programm um das Inverse zu finden heißt `MInv`:

```
>java MInv 5 13
```

6.3.5 Einfach & Schwierig

Es gibt nun zwei Eigenschaften, die die Primzahlverfahren sehr sicher machen:

a) Die Zerlegung von großen Zahlen in ihre Primfaktoren ist schwierig, die Multiplikation dagegen ist einfach:

Sind p und q zwei 300-stellige Primzahlen, so ist $p*q$ einfach zu berechnen; die Primfaktorzerlegung einer 600-stelligen Zahl hingegen ist sehr zeitaufwändig ("pröbeln"). Genau diese Schwierigkeit nutzt das RSA Verfahren.

b) Exponenten Modulo einer Primzahl zu rechnen ist einfach. Die Umkehrung (den sog. diskreten Logarithmus) zu finden ist schwierig. Diese Schwierigkeit wird vom Diffie/Hellman-Verfahren wie auch vom ElGamal-Algorithmus ausgenutzt:

$a^b \bmod p$ zu berechnen ist einfach (siehe Beispiel 6).

Aus der Gleichung $a^n \bmod p = s$ das n zu berechnen, ist hingegen schwierig.

6.3.6 Hash-Funktionen

Hash-Funktionen sind schnell zu berechnende Schlüsseltransformationen. Sie bilden Daten (Suchschlüssel, Sortierschlüssel, beliebige Objekte, E-Mails, ...) auf einen vergleichsweise kleinen Wertebereich ab. So wird aus einem großen Objekt eine kleine Datenstruktur aus wenigen Bytes.

Hash-Funktionen haben die folgenden Eigenschaften:

schnell berechenbar	Eine Hash-Funktion soll sehr schnell berechnet werden können.
kleiner Wertebereich	Objekte beliebiger Größe werden auf wenige Bytes abgebildet. JAVA verwendet eine Hash-Funktion für Strings, die jede Zeichenkette auf lediglich 4 Bytes abbildet.
nicht umkehrbar	Hash Funktionen können nicht rückgängig gemacht werden. Es handelt sich hier um eine Art Einwegfunktion. Jedoch nicht so, dass die Funktion sehr schwierig umzukehren ist, wie dies bei den Verschlüsselungsverfahren der Fall ist, sondern, dass die Funktion überhaupt nicht umzukehren ist. Aus dem Hash-Wert können die ursprünglichen Daten nicht wieder rekonstruiert werden. Das ist übrigens eine einfache Konsequenz aus obiger Tatsache des *kleinen Wertebereiches*. Mehrere Objekte können denselben Hash-Wert erhalten.
gute Streuung	Hash-Funktionen sollen im Wertebereich stark streuen. Das heißt: Zwei unterschiedliche Objekte sollten mit großer Wahrscheinlichkeit zwei verschiedene Hash-Werte liefern.

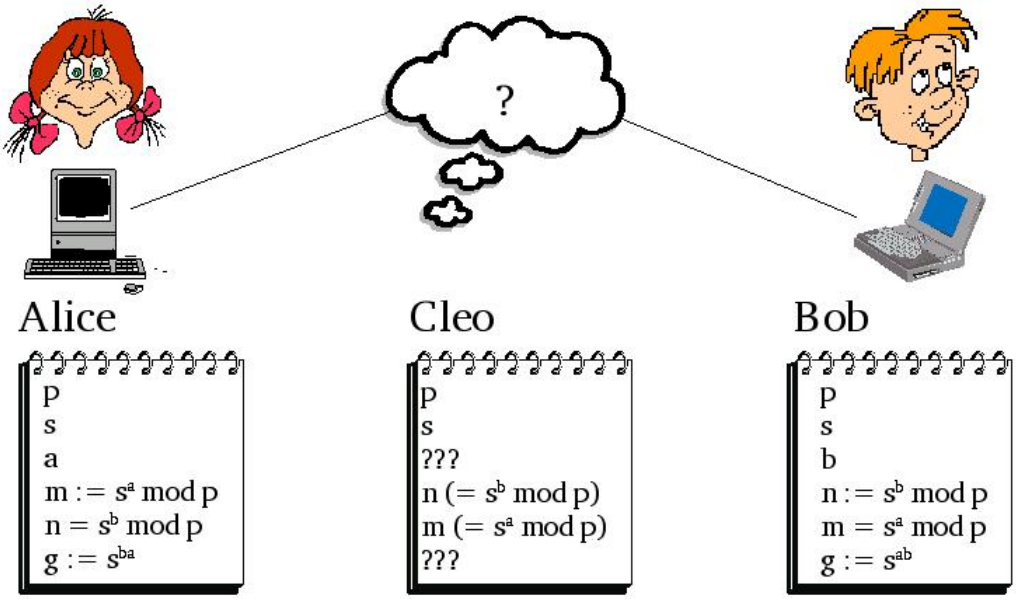
Einsatzgebiete:

- Digitale Signatur (siehe auch Kapitel 6.2.5 auf Seite 26)
- Digitaler Fingerabdruck
- Passwortlisten (siehe auch Kapitel 5.1.2 auf Seite 21)
- Hash-Tabellen (nicht Inhalt dieses Kurses.)

Beispiele:

- Der JAVA Hash-Code von “Geheimnachricht” ist 0x4A70B2C1.
- Der JAVA Hash-Code von “Geheim Nachricht” ist 0xAB60729F.

6.4 Das Verfahren von Diffie/Hellman



Das Verfahren von Diffie/Hellman (DH-Verfahren[21] 1976) erlaubt es, Daten verschlüsselt zu übermitteln, ohne vorab einen geheimen Schlüssel auf einem separaten Kanal zu transportieren[22].

6.4.1 Ausgangslage

- Ausgangssituation: Alice (A) und Bob (B) wollen Nachrichten über eine öffentlich zugängliche Leitung austauschen. Diese Leitung wird möglicherweise abgehorcht.
- Alice und Bob hatten vorher noch keine Schlüssel miteinander ausgetauscht.
- Das Diffie/Hellman-Verfahren erlaubt es, eine Botschaft über eine Leitung zu übermitteln, ohne dass ein "Horcher" die Nachricht verstehen kann.
  Der Horcher sei im Folgenden Cleo (C).
- Es wird davon ausgegangen, dass der "Horcher" sich nicht als Sender oder Empfänger ausgeben kann. (Die Transaktion sei zu schnell, als dass die *Man-in-the-Middle*-Attacke funktionieren könnte.)

[21] Whitfield Diffie, Martin Hellman und Ralph Merkle
[22] Das Verfahren wurde schon vor Diffie und Hellman von Malcolm Williamson vorgeschlagen.

6.4.2 Vorgehen im Diffie/Hellman-Verfahren

Das Diffie/Hellman-Verfahren besteht aus folgenden Schritten:

1. Die Kommunikationspartner A und B (Alice und Bob) entscheiden sich gemeinsam für eine große Primzahl p. (Je größer die Primzahl, umso sicherer das Verfahren.)

2. A und B suchen eine Zahl s, die kleiner als p ist (Bedingung: $1 < s < p$).[23] Wichtig: Die beiden Zahlen p und s können unverschlüsselt über die Telefonleitung übermittelt werden; sie sind also öffentlich zugänglich. Cleo (C) hört p und s, kann aber mit diesen Zahlen noch nichts anfangen.

3. Alice sucht sich im Geheimen eine Zahl a.
   Bob sucht sich im Geheimen eine Zahl b.
   Die beiden Zahlen sollten kleiner als $p - 1$ sein.

4. A berechnet $m := s^a \bmod p$ und schickt das Resultat an B.
   B berechnet $n := s^b \bmod p$ und schickt das Resultat an A.
   Cleo hört zwar m und n mit und kann damit nun theoretisch a und b berechnen. Er braucht dazu aber viel zu lange.

5. A berechnet die Geheimzahl $g = n^a \bmod p$
   B berechnet die Geheimzahl $g = m^b \bmod p$
   Bem. $g = n^a = m^b$, denn $n^a = (s^b)^a = s^{ba} = s^{ab} = (s^a)^b = m^b$
   C kann g nicht in nützlicher Frist berechnen.
   Das einzige mathematische Verfahren für C, um g zu "berechnen", heißt *ausprobieren*[24].

6. A und B können nun ihre Nachrichten mit g verschlüsseln und entschlüsseln (z. B. mit dem XOR-Verfahren (siehe auch Kapitel 6.2.3 auf Seite 25)).

6.4.3 Ein Zahlenbeispiel

1. Alice und Bob bestimmen gemeinsam die folgenden Zahlen und tauschen diese übers Internet aus:

$$\begin{aligned} p &= 467 \\ s &= 44 \end{aligned}$$

2. Alice wählt sich eine beliebige Geheimzahl a, Bob wählt sich die Geheimzahl b. Diese Zahl darf nicht übers Netz verraten werden! Auch nicht gegenseitig. Alice wird b nicht erfahren; ebenso wird Bob die Zahl a nie erfahren:

[23] Genaugenommen sollte s eine sogenannte Primitivwurzel modulo p sein (Siehe [Buchmann 2001]). Das Verfahren funktioniert auch für andere Zahlen, kann aber unter Umständen einfach *geknackt* werden.

[24] Es gibt einige mathematische Tricks (das Babystep-Giantstep-Verfahren von Shanks, der Pollard-ρ-Algorithmus oder das Verfahren von Pohlig-Hellman), um schneller zum Ziel zu kommen, und es gibt Fälle von *einfachen* Primzahlen, bei denen das Verfahren schneller *geknackt* werden kann als stures ausprobieren; vgl. [Buchmann 2001].

$$\begin{aligned} a &= 101 \\ b &= 133 \end{aligned}$$

3. Alice berechnet $m = s^a \bmod p = 44^{101} \bmod 467 = 292$. Alice schickt $m = 292$ an Bob.

   Bob berechnet $n = s^b \bmod p = 44^{133} \bmod 467 = 115$. Bob schickt $n = 115$ an Alice.

4. Alice berechnet $g = n^a \bmod p = 115^{101} \bmod 467 = 150$.

   Bob berechnet $g = m^b \bmod p = 292^{133} \bmod 467 = 150$.

   Oh Wunder, es gibt dieselbe Geheimzahl. Nur Alice und Bob kennen sie!

5. Jetzt können Alice und Bob die Zahl 150 als Basis für ein einfaches Verschlüsselungsverfahren benutzen (z. B. das XOR-Verfahren).

6.5 Das Verfahren von Rivest, Shamir und Adleman (RSA)

- Das RSA-Verfahren wurde von Ronald R. Rivest, Adi Shamir und Leonard M. Adleman entwickelt.
- Das RSA-Verfahren ist ein "Public Key Verfahren". Das heißt, der Algorithmus und der Schlüssel zum Verschlüsseln (codieren, chiffrieren) von Botschaften wird öffentlich bekannt gegeben. Nur der Schlüssel zum Entschlüsseln (decodieren, in Klartext zurückverwandeln) der Botschaften wird geheim gehalten.
- Das RSA-Verfahren basiert auf der Tatsache, dass es sehr einfach ist, zwei Primzahlen miteinander zu multiplizieren, dass es aber äußerst aufwändig ist, die beiden Primzahlen wieder zu finden, falls nur noch das Produkt bekannt ist.

6.5.1 Vorgehen im RSA-Verfahren

Das RSA-Verfahren besteht aus vier Schritten.

1. Der Empfänger generiert ein Schlüsselpaar: einen privaten (geheimen) und einen dazu passenden öffentlichen Schlüssel,
2. der Empfänger publiziert den öffentlichen Teil,
3. der Sender verschlüsselt mit dem öffentlichen Schlüssel des Empfängers seine Botschaft und
4. der Empfänger entschlüsselt die Botschaft mit seinem geheimen Schlüssel.

Der Trick dabei ist, dass nur der Empfänger die Botschaft in sinnvoller Zeit entschlüsseln kann, denn nur er kennt die Primfaktorzerlegung des Schlüssels, wie wir gleich sehen werden. Nun aber die 4 Schritte im Detail:

Schritt 1: Schlüssel generieren In diesem Schritt generiert der Empfänger einen öffentlichen Schlüssel. Alle können danach mit diesem Schlüssel Botschaften chiffrieren (verschlüsseln), aber nur der Empfänger kann sie wieder dechiffrieren (entschlüsseln).

a) Der Empfänger sucht zwei (möglichst große) Primzahlen p und q.

b) Der Empfänger berechnet $r = p \cdot q$.

c) Der Empfänger berechnet zudem $s = (p-1) \cdot (q-1)$.

d) Der Empfänger bestimmt ein beliebiges c mit den beiden folgenden Eigenschaften: $c < s$ und $\text{ggT}(c, s) = 1$. Das erreicht der Empfänger zum Beispiel einfach, indem er eine Primzahl sucht, die kleiner als s ist. Hier kann das JAVA Programm `GCD` eingesetzt werden:

```
>java GCD c s
```

muss 1 ergeben!

Schritt 2: Veröffentlichen des Schlüssels Der Empfänger gibt r und c als öffentlichen Schlüssel bekannt. Zum Beispiel steht auf seiner Homepage vereinfacht: *Der öffentliche Schlüssel von gerd.gesell@tbz.ch ist (r = 289073 und c = 353).*

Schritt 3: Verschlüsseln einer Botschaft

a) Der Sender zerhackt seine Botschaft in kleinere Stücke, die danach einzeln verschlüsselt werden. Sind die Stücke sehr klein, z. B. 8 Bit, so kommt das Verfahren fast einer monoalphabetischen Substitution gleich. Die Stücke dürfen jedoch nicht mehr Bits in Anspruch nehmen als die Zahl r. Ist z. B. $r = 134565$, so darf ein Stück nicht mehr als 17 Bit in Anspruch nehmen ($2^{17} = 131072$). Der Einfachheit halber würde man in einem solchen Fall wohl 2 Byte belegen (2 Byte = 16 Bit < 17 Bit).

In mathematischen Worten würde man wohl sagen: Die Originalbotschaft wird in Stücke B_i der Bitlänge des (ganzzahligen) Zweierlogarithmus von r unterteilt.

b) Diese "Kurzinformationen" B_i muss der Sender nun mit r und c verschlüsseln:

$$\begin{aligned} G_1 &= B_1^c \bmod r \\ G_2 &= B_2^c \bmod r \\ &\ldots \end{aligned}$$

Verwenden Sie das JAVA Hilfsprogramm

`>java AhBmC Bi c r`.

c) Diese "Chiffrate" (G_1, G_2, ...) sendet er/sie an den Empfänger. Das kann problemlos über die öffentliche Telefonleitung geschehen, denn nur der Empfänger kann diese entschlüsseln.

Schritt 4: Entschlüsseln der Botschaften

a) Der Empfänger berechnet $d := c^{-1} \bmod s$.

Diesen Dechiffrierexponenten d kann nur er berechnen, denn nur er kennt s!

Bem.: Alle können zwar rein theoretisch r in die Primfaktoren p und q zerlegen und so auch s berechnen; aber bei großen Primzahlen p und q ist dieser Aufwand immens.

Verwenden Sie, um d zu berechnen, das JAVA Programm

`>java Minv c s`.

b) Alle Chiffrate (G_1, G_2, ...) kann der Empfänger mit

$$\begin{aligned} B_1 &= G_1^d \bmod r \\ B_2 &= G_2^d \bmod r \\ &... \end{aligned}$$

nun entschlüsseln. Verwenden Sie wiederum das JAVA Programm

`>java AhBmC Gi d r`.

6.5.2 Ein Zahlenbeispiel

Schritt 1: Empfänger Wertzuweisungen:

$$\begin{aligned} p &= 467 \\ q &= 619 \\ r &= 467 \cdot 619 = 289073 \\ s &= 466 \cdot 618 = 287988 \\ c &= 353 \end{aligned}$$

Schritt 2: Veröffentlichen Öffentlich bekannt geben: $r = 289073$, $c = 353$

Schritt 3: Sender Die Botschaft sei "Hallo". Die Botschaft wird in Stücke der Länge 2 Byte (= 16 Bit) unterteilt. (Die "x" steht für *hexadezimale Codierung.*)

$$\begin{array}{rcrcrcr} B_1 & = & \text{“Ha”} & = & x4861 & = & 18592 \\ B_2 & = & \text{“ll”} & = & x6C6C & = & 27756 \\ B_3 & = & \text{“o”} & = & x6F & = & 111 \end{array}$$

$$\begin{array}{rcrcrcr} G_1 & = & B_1^c \bmod r & = & 18529^{353} \bmod 289073 & = & 200883 \\ G_2 & = & B_2^c \bmod r & = & 27756^{353} \bmod 289073 & = & 212601 \\ G_3 & = & B_3^c \bmod r & = & 111^{353} \bmod 289073 & = & 12789 \end{array}$$

Übermittelt wird nun die chiffrierte Botschaft :

$(G_1, G_2, G_3) = (200883, 212601, 12789)$

Schritt 4: Empfänger Der Empfänger berechnet zunächst d, den sog. Dechiffrierexponenten.

$$d = c^{-1} \bmod s = 206405$$

$$\begin{array}{rcrcrcrcr} B_1 & = & 200883^d \bmod r & = & 18529 & = & x4861 & = & \text{“Ha”} \\ B_2 & = & 212601^d \bmod r & = & 27756 & = & x6C6C & = & \text{“ll”} \\ B_3 & = & 12789^d \bmod r & = & 111 & = & x6F & = & \text{“o”} \end{array}$$

6.6 Der eigene öffentliche Schlüssel

Wichtig bei öffentlich benutzten Verfahren ist es natürlich, dass die Zahlen, Schlüssel, Hash-Codes etc. in einem standardisierten Protokoll übermittelt werden.

Mit PGP (pretty good privacy: `www.pgpi.org`) von Phil Zimmermann existiert eine standardisierte Implementierung des Public Key Verfahrens. Unter Linux wird auch GnuPG (`www.gnupg.org`) als Open Source Version angeboten.

PGP arbeitet mit einem Vertrauens-Netzwerk. Jedem Schlüssel in meinem Schlüsselbund kann eine Vertrauensstärke angegeben werden. So gibt es Schlüssel, deren Herkunft ich eher traue als anderen. Wenn ich nun einen neuen Schlüssel in meinen Schlüsselbund aufnehme, so schaue ich nach, ob jemand, dem ich traue, diesem Schlüssel bereits vertraut: In solchen Fällen kann ich dem neuen Schlüssel auch eher trauen.

6.7 JAVA Hilfsprogramme

Holen Sie sich die mathematischen Hilfsprogramme vom BSCW und entpacken Sie diese auf Ihrem lokalen Rechner: XorKryptRandom.zip und math.zip

6.7.1 `XorKryptRandom`

Das Programm `XorKryptRandom` hat den Zweck, eine mit dem XOR-Verfahren geheim gehaltene Botschaft zu entschlüsseln oder auch zu verschlüsseln. Da es sich um ein symmetrisches Verfahren handelt, kann für beide Schritte (verschlüsseln, entschlüsseln) dasselbe Programm eingesetzt werden.

Dem Programm wird beim Starten ein Startwert (*Seed*) für den Zufallszahlengenerator mitgegeben. Mit diesem *Seed* wird eine zufällige Bytefolge generiert. Diese Folge wird *bitweise* mit XOR mit dem Originaltext verknüpft. Das Resultat ist die verschlüsselte Botschaft.

Beispiel 7 (XorKryptRandom) *Sei* `original.txt` *die zu verschlüsselnde Botschaft. Der Startwert für den Zufallszahlenalgorithmus wird zufällig gewählt, muss aber beiden Parteien (Sender, Empfänger) bekannt sein; z. B. 56. Gehen Sie wie folgt vor, um die Botschaft* `original.txt` *in eine Datei* `krypt.cpt` *zu verschlüsseln:*

```
>java XorKryptRandom 56 original.txt  >krypt.cpt
```

entschlüsseln:

```
>java XorKryptRandom 56 krypt.cpt  >original.txt
```

6.7.2 Größter gemeinsamer Teiler: `GCD`

Mit dem Programm `GCD` (greatest common divisor = größter gemeinsamer Teiler) wird der *ggt()* von zwei Zahlen berechnet.

Beispiel 8 (ggt() von 38 und 57) `>java GCD 38 57`

6.7.3 Das Inverse modulo einer Primzahl: `MInv`

Mit dem Programm `MInv` (Modulo-Inverses) wird das Inverse modulo einer Primzahl p berechnet (siehe auch Kapitel 6.3.4 auf Seite 28).

Beispiel 9 (Inverses von $5(\text{mod}13)$**)** `>java MInv 5 13`

Ergebnis: ***8****.*

6.7.4 Potenzieren modulo einer Primzahl: `AhBmC`

Mit dem Programm `AhBmC` *"a hoch b* mod *c"* wird eine Zahl a b Mal mit sich selbst multipliziert. Danach wird der Divisionsrest mod c berechnet.

Beispiel 10 (a^b mod $c : 43^{11}$ mod 13**)** `>java AhBmC 43 11 13`

Ergebnis: ***10****.*

7 Übungen und Aufgaben

7.1 Warenkorb

Installieren Sie den Demo-Warenkorb. Gehen Sie wie folgt vor:

1. Download: Je nach Infrastruktur stellt Ihnen die Lehrperson ein ZIP-File mit PHP-Quellcode zur Verfügung. Extrahieren Sie die Dateien auf Ihrem PC und installieren Sie die Applikation entsprechend.
2. Kaufen Sie eine Ware (`http://localhost/wako/produkte.php`) und vergleichen Sie danach die Bestellungen (`http://localhost/wako/admin.php?passwort=123456`).
3. Versuchen Sie eine Session Ihres Nachbarn zu *hijacken*. Das geht hier ganz ohne Kenntnisse von TCP.
4. Kaufen Sie einen Artikel, der nicht auf der Artikelliste zu finden ist. Spionieren Sie als Kunde in der Datenbank: GET-Parameter ausprobieren oder systematisch abholen.
5. Betreiben Sie "E-Shop-Lifting": Kaufen Sie einen Artikel zu verfälschtem Preis.
6. Fügen Sie neue Produkte in die Produkteliste ein, und testen Sie eine Bestellung.

7.1.1 Alternative: Neuer Shop

Alternativ zum Anpassen des bestehenden Shops, kann unter Umständen auch ein ganz neuer Shop programmiert werden. Dieser muss natürlich mindestens so viele (aber fehlerfreie) Funktionen bieten, wie der Demo-Warenkorb.

- Der eigene neue Web-Shop muss mindestens folgende Funktionen enthalten:
  - Warenkorb mit Total-Anzeige,
  - Artikelauswahl (Liste oder Suche) und
  - eine Bestellmöglichkeit.
- Der neue Shop muss mindestens zwei Verbesserungen gegenüber dem Demo-Warenkorb enthalten.
- Vergessen Sie nicht, die Testfälle **vorab** zu beschreiben.

7.2 Shop-Vergleich

Es gibt eine ganze Menge an bestehenden Webshops - allerdings von unterschiedlicher Qualität. Untersuchen Sie möglichst Ihnen bereits bekannte Webshops. Diese müssen nicht optimal sein - im Gegenteil. Sie sollen ja gerade Schwachstellen entdecken. Analysieren Sie mindestens 2 Shops. Hier sind noch ein paar Kriterien aufgelistet, aber Sie können eigene hinzufügen und müssen die aufgelisteten nicht unbedingt verwenden.

- Warenkorb
- Suchfunktionen
- Zahlungsverkehr / -möglichkeiten
- Kategorisierung von Artikeln
- Konfiguration von Artikeln
- Benutzerschnittstelle (Bestellvorgehen, Ergonomie; Verfolgung Auftragsstatus, Usability)
- Login und Benutzerdaten
- Undo-Funktionalität (Rückgängig machen von Aktionen)
- Zusätzliche Funktionen

In einigen Shops ist es auch möglich etwas zu den folgenden Punkten zu sagen:

- eingesetzte Architektur (ASP, JSP, PHP, Java-Script, Applets, ...)
- Sicherheit (GET-Parameter, `https`, ...)
- verwendete Standardsoftware oder Frameworks

Beispiele für mögliche Shops (aber suchen Sie auch eigene):

- `www.amazon.de`
- `www.books.ch`
- `www.buch.ch`
- `www.bahn.de`
- `www.sbb.ch`
- `www.directmedia.ch`
- `www.1a-dvdshop.ch`
- `www.cede.ch`
- `www.laserzone.ch`
- `www.emp.de`
- `www.soundaid.ch`
- `www.dvd-shop.ch`
- `www.jukebox.ch`
- `www.alphamusic.de`

Erstellen Sie ein Dokument, das Ihre Ergebnisse darstellt und geben dies auf dem BSCW ab. Der genaue Abgabeort und -termin wird Ihnen separat bekanntgegeben.

7.3 Sicherheit

7.3.1 Schutz gegen Angriffe

Geben Sie zu vier der folgenden möglichen Angriffe aus dem Kapitel Sicherheit (siehe auch Kapitel 5.1 auf Seite 21) eine mögliche Abhilfe.

Spionage mittels GET-Parameter	....
Passwort Cracker und Guesser	....
Horcher, Man-in-the-Middle	....
E-Shop Lifting	....
Viren, Würmer, Enten	....
Trojanische Pferde	....
DoS Attacken	....

7.3.2 Angriff

Beschreiben Sie, wie Sie als *Hacker* vorgehen würden, um Sicherheitslöcher im eBanking auszunutzen. Was ist zu tun, um in das System einzudringen, dieses auszuhorchen oder zu manipulieren.

a) Passwort auf PC gespeichert	....
b) Passwort nicht geändert	....
c) Alte (unsichere) Browserversion	....
d) Cache nicht geleert	....
e) Kein Virenschutz	....
f) “Fake” E-Mails beantwortet	....
g) User gibt telefonisch Passwörter durch	....
h) User achtet nicht auf sichere Verbindung	....

7.4 Verschlüsselung

Voraussetzungen:

- Sie haben einen Internetzugang und einen gängigen Web-Browser.
- Sie haben die Möglichkeit, auf Ihrem Computer Software (JSDK, PGP, GnuPG usw.) zu installieren.
- Sie sind in der Lage, Text zu editieren.
- Sie sind in der Lage, E-Mails zu versenden.

1. Entpacken Sie die folgenden Dateien auf Ihrem PC:

   (siehe auch Kapitel 6.7 auf Seite 36)

2. Schreiben Sie die Lösung zu den drei folgenden Aufgaben in ein elektronisches Dokument:

   - Denken Sie sich in das XOR-Verfahren mit einem Random-Seed ein und versuchen Sie, den bereitgestellten Text aus **XorKryptRandom (WelcheLebensform.cpt)** zu "knacken". Schauen Sie dazu im Kapitel XOR (siehe auch Kapitel 6.2.3 auf Seite 25), im Kapitel `XorKryptRandom` (siehe auch Kapitel 6.7.1 auf Seite 36) und in der heruntergeladenen `README.TXT`-Datei nach. Der Zufallszahlengenerator von JAVA wurde verwendet. (PS: Der gewählte Seed liegt zwischen 0 und 100)
     - Wie lautet die gesuchte Lebensform?
     - Beschreiben Sie kurz Ihr Vorgehen.
   - Beschreiben Sie in wenigen Sätzen (100-200 Wörter), wie ein längerer Text, der mittels *monoalphabetischer Substitution* (siehe auch Kapitel 6.2.2 auf Seite 25) verschlüsselt ist, geknackt werden kann. Geben Sie alle allfälligen Quellen (auch Internet) an.
   - Beschreiben Sie (100-200 Wörter), wie mit dem Verfahren öffentlicher Schlüssel (Public-Key) ein Text verschlüsselt wird und wieder gelesen werden kann. Gehen Sie nicht auf Primzahlverfahren oder spezielle Schlüssel (ElGamal, RSA, Diffie/Hellman , DES, ...) ein, sondern erklären Sie, wer zu welchem Zeitpunkt mit welchem Schlüssel (oder Teil davon) was tun muss. Beschreiben Sie auch, wie eine digitale Signatur (Unterschrift) mit Public-Key-Verfahren funktioniert: Wessen Schlüssel wird wann eingesetzt?

3. Die drei Texte (XOR, monoalphabetische Substitution, Public Key) sind elektronisch zu verfassen.

7.5 Verschlüsselung - Praxis

Das Ziel dieser Aufgabe ist es ein oder 2 Verschlüsselungstools mit asymmetrischer Verschlüsselung einzusetzen und die Arbeitsweise und einige Details kennen zu lernen. Zum Beispiel PGP oder GnuPG.
Auf dem BSCW liegt im Ordner e-learning II mein Public-Key. Ihr müsst mir eine verschlüsselte Mail senden, die mit diesem Schlüssel verschlüsselt ist. Wer nur mit online-Mail arbeitet, muss eine Datei oder die Zwischenablage verschlüsseln und mir den verschlüsselten Code (Cipher-Text) als Anhang senden. Zusätzlich müssen Sie sich ein Schlüsselpaar generieren und den Public-Key veröffentlichen (BSCW, per Mailanhang oder auf einem öffentlichen Schlüsselserver). Damit verschlüssele ich eine Nachricht und sende Sie Euch zurück. Diese enthält eine Information, die Sie entschlüsseln müssen.

1. Generieren Sie sich mittels PGP oder GnuPG ein *Public/Private-Key-Paar* zu **Ihrer** E-Mailadresse.

   Legen Sie Ihren Public-Key auf dem BSCW ab [25]

2. Holen Sie den Public-Key vom BSCW und legen diesen in Ihrem *Key-Ring* ab. Je nach System ist dem Schlüssel noch ein "trust-level" anzugeben.

3. Verschlüsseln Sie Ihren Text oder Ihr Mail und senden Sie dieses in einem verschlüsselten und digital signierten E-Mail an `gerd.gesell@bluewin.ch`.

[25] Sie können Ihren Schlüssel auch auf einem Key-Server publizieren.

7.6 Standard Web-Shops

Installation eines *Open Source* Standard Web-Shops und dessen Konfiguration (Customizing) oder programmatische Änderung.

- Abklären der Bedürfnisse
- Beschaffen der Dateien
  - osCommerce (`http://www.oscommerce.org`)
  - phPepperShop (`http://www.phpeppershop.com`)
  - Intrexx (`http://www.intrexx.com`)
  - Interchange (`http://www.icdevgroup.org`)
  - pgMarket (`http://www.pgmarket.net`)
  - phPay (`http://www.phpay.de`)
  - phpShop (`http://www.phpshop.org`)
  - Magento `http://www.magentocommerce.com/de/`
  - OXID eSales `http://www.oxid-esales.com/de/startseite`
  - und so weiterrrrr.....
- Teilweise finden Sie bereits gute Vergleichsberichte im Internet. Zum Beispiel: http://www.shopanbieter.de/news/archives/2132-Drei-Open-Source-Shopsysteme-im-Vergleich.html
- Schrittweise Dokumentation der Installation als Anleitung. Die Dokumentation ist so durchzuführen, dass der Shop später jederzeit mit geringem Aufwand wieder installiert werden kann.
- Änderung planen (Konfiguration oder Programmänderung)
- Änderung durchführen
- Änderung dokumentieren und testen.

8 GnuPG

Wie wird `GnuPG` von Hand eingesetzt? Hier einige wichtige Befehle und Optionen, falls Sie sich entscheiden, nicht die graphische Benutzeroberfläche von PGP zu verwenden.

8.1 Installation

Zuerst müssen Sie das gpg-Tool (z. B. `gpg.exe`) herunterladen.
Zum Beispiel hier: (`www.pgpi.org` oder `www.gnupg.org`)

1. Erstellen Sie ein Verzeichnis `c:\temp\gpg`[26] und kopieren Sie alles vom Netz dahinein.
2. Wechseln Sie entweder ins oben erstellte Verzeichnis (`cd c:\temp\gpg`) oder geben Sie dieses Ihrem System-Pfad bekannt.

8.2 GPG-Home Verzeichnis

GnuPG benötigt ein Heim-Verzeichnis, um die Schlüssel zu speichern. Wenn nichts angegeben wird, sucht sich GnuPG selbständig ein solches Verzeichnis aus. Wenn Sie auf den Maschinen nicht überall Schreibrechte haben oder wenn Sie das Verzeichnis explizit angeben wollen, verwenden Sie bei jedem Aufruf von `gpg` die Option `-homedir <Verzeichnis>`. Dabei steht `<Verzeichnis>` für das von Ihnen gewählte Verzeichnis.

8.3 Schlüssel generieren

Mit der Option `-gen-key` wird ein Schlüssel generiert. ElGamal-Schlüssel sind in der Regel sicher. Wegen einem Implementationsfehler sollten Sie aber nicht zum Unterschreiben eingesetzt werden. Für unsere kleine Übung ist das jedoch kein Problem, denn der Aufwand, den privaten Schlüssel dennoch zu "knacken", ist immer noch hoch genug.

```
gpg --homedir  c:\temp\gpg --gen-key
```

8.4 Importieren von Schlüsseln

Damit Sie später für jemanden etwas verschlüsseln können, müssen Sie seinen Public-Key importieren; im Volksmund gesagt: Hängen Sie den öffentlichen Schlüssel des Empfängers an Ihren Schlüsselbund (`Key-Ring`). Das geschieht mit der Option `import`.

```
gpg --homedir c:\temp\gpg --import phi.key
```

[26] Falls Sie in beliebigen Verzeichnissen Schreibrechte haben, können Sie selbstverständlich das Programm auch in ein anderes Verzeichnis kopieren. Sie können dann auch auf den (im Folgenden immer erwähnten Zusatz "`homedir`") verzichten.

Den erfolgreichen Import testen Sie am einfachsten mit dem Auflisten aller Schlüssel an Ihrem "Schlüsselbund".

```
gpg --homedir c:\temp\gpg --list-keys
```

8.5 Schlüssel unterschreiben und beglaubigen

Es gibt noch zwei weitere Befehle, die je nach Anwendung der Schlüssel einzusetzen sind: Schlüssel unterschreiben (`-sign-key`) und Schlüssel beglaubigen (`-edit-key` und `trust`).

8.6 Verschlüsseln / Entschlüsseln einer Botschaft

Das Verschlüsseln für einen Empfänger (z. B. `phi@gressly.ch`) ist nun, nachdem die obigen Schritte durchgeführt sind, keine Hexerei mehr. Der Befehl `-encrypt` verschlüsselt den Text. Hier können mehrere Empfänger angegeben werden. Sobald eine Leerzeile eingegeben wird, wird der Text für alle angegebenen Empfänger verschlüsselt.

```
gpg --homedir c:\temp\gpg --encrypt geheim.txt
```

Zum Entschlüsseln verwenden Sie den Befehl `-decrypt`.

Notizen:

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

A Verzeichnisse

Literatur

[Amor 2001] D. Amor: *Die E-Business-(R)Evolution* [Galileo Press GmbH], 2001 (ISBN:3-89842-185-6)

[Buchmann 2001] J. Buchmann: *Einführung in die Kryptographie* [Springer] 2001 (ISBN: 3-540-41283-2)

[CT Nr. 26 2002] Heise Verlag, CT-Magazin, Nr. 26 vom 16. Dez. 2002. "Virtueller Ladendiebstahl" Seite 92

[iX Nr. 10 2003] Heise Verlag, iX Magazin für professionelle Informationstechnik, Nr. 10: Oktober 2003 "Sonderangebote" Seite 62

[Jacobsen 2005] Jens Jacobsen: *Website-Konzeption* [Addison-Wesley Verlag] 3. erweiterte Auflage 2005; ISBN: 3-8273-2249-9

[Merz 2001] H. Merz: *Praxis-Lexikon e-Business* [verlag moderne industrie] 2003 (ISBN: 3-478-24940-6)

[Meier/Stormer 2008] Andreas Meier, Henrik Stormer: *eBusiness und eCommerce* [Springer Verlag] 2008 ISBN: 9-783540-850168

[Modul 150] *Modulidentifikation M150* [i-ch] 2006

[Staud 2005] Josef L. Staud: *Datenmodellierung und Datenbankentwurf* [Springer Verlag] 2005 ISBN: 3-540-20577-2

[Uhr 2003] W. Uhr: *E-Business* [http://www.tu-dresden.de] 2003

[Wahrig] *WAHRIG* Deutsches Wörterbuch

[Zwerger/Paulus 2002] F. Zwerger, S. Paulus: *E-Business Projekte* [Galileo Business] 2003 (ISBN: 3898421953)

Abbildungsverzeichnis

B Index