# IX ASPEKTE DER IT-KOMMUNIKATION

S. BARTELS, 30.7.2018

## 1. Aufbau und Organisation des Internets

Das Internet stellt die Basis der digitalen Welt und der damit verbundenen Kommunikation in der Informationstechnologie dar. Seine Aufgabe ist der Transport von Daten und diesem Zweck dienen eine technische Infrastruktur, die unter anderem aus Glasfaserkabeln, Routern und Netzwerkservern besteht, sowie Konzepte der Datenübermittlung. Der Austausch von Daten funktioniert dabei unabhängig von vorhandenen Geräten und (bisher) der Art der Daten. Die technische Infrastruktur setzt sich aus zahlreichen Einzelnetzwerken wie denen von Internet-Providern sowie Firmen und Universitäten zusammen, die jeweils mit Knotenpunkten verbunden sind. Von diesen Knotenpunkten existieren weltweit ca. 340, die untereinander vernetzt sind. Ihre Vernetzung garantiert, dass zwischen zwei Knotenpunkten mindestens zwei unabhängige Verbindungen existieren, und so eine hohe Ausfallsicherheit erreicht wird.
Der Austausch von Daten erfolgt über ein Internet-Protokoll (IP), welches ein plattform- und anwendungsunabhängiges Datenformat definiert. Informationen wie E-Mails, Inhalte von Internetseiten oder Videokonferenzen werden dabei in kleinere Datenpakete zerlegt und mittels eindeutiger IP-Adressen versendet. Die Datenpakete können bis zu 65.000 Byte groß sein und bestehen aus einem Kopfbereich, dem sogenannten *Header*, mit Absende- und Ziel-IP-Adresse und einem Nutzdatenbereich, der lediglich 15.000 Byte groß ist. Mittels *Routern* werden die Einzelpakete im Internet verschickt. Sie können dabei unterschiedliche Wege nehmen und kommen möglicherweise unsortiert beim Empfänger an. Durch die Informationen im Header können sie jedoch eindeutig zusammengefügt werden.

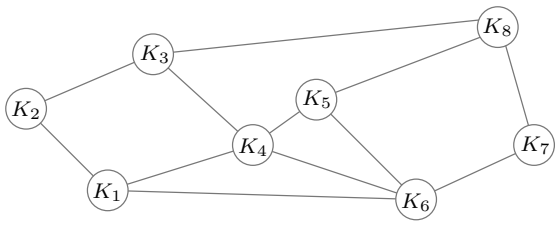


Abbildung 1. Schematische Darstellung der Vernetzung der Knotenpunkte des Internets.

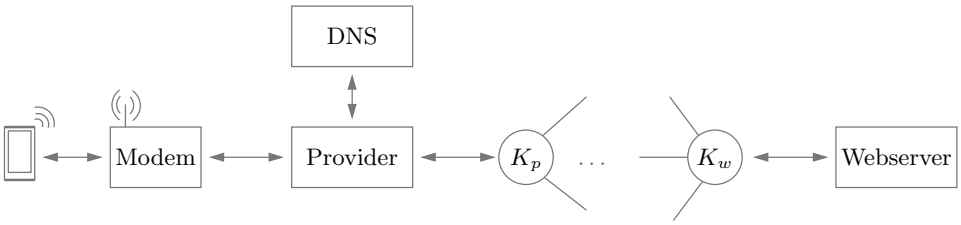


ABBILDUNG 2. Kommunikationspunkte beim Aufruf einer Internetseite durch ein Endgerät über Knotenpunkte $K_p$ und $K_w$ des Internets und Ermittlung der IP-Adresse über einen Domain Name Server (DNS).

**Beispiel 1.1.** *Wir betrachten den Aufruf einer Internetseite durch ein internetfähiges Gerät wie ein Smartphone oder ein Laptop. Das Gerät erhält vom Modem, das auch als Router fungiert, eines lokalen Netzwerks (oder LAN für* local area network*) eine interne IP-Adresse. Beim Aufruf einer Internetseite wird die Anfrage zusammen mit der internen IP-Adresse an das Modem geschickt, welches über eine Schnittstelle (auch als* Proxy *bezeichnet) eine Verbindung zum* Internet Service Provider *(ISP) herstellt. Das Modem hat eine vom Provider zugeteilte und unter Vorratsdatenspeicherung temporär registrierte dynamische IP-Adresse, während der Provider eine statische besitzt. Der Provider ermittelt über einen* Domain Name Server *(DNS) die IP-Adresse der angewählten Seite. Anschließend wird die Anfrage an den Server der Seite verschickt, welcher daraufhin die gewünschten Daten an das anfragende Gerät in Form kleiner Datenpakete versendet. Mit einer Kapselung gelangen die Pakete über den Provider und das Modem zum Endgerät, welches die Anfrage abgesendet hat.*

**Bemerkungen 1.2.** *(i) Aufgrund von Weiterentwicklungen der Lichtsignaltechnologie werden nur ca. 3% der Kapazitäten der verfügbaren Glasfaserkabel genutzt.*
*(ii) In Deutschland verursacht der Stromverbrauch des Internets durch Endgeräte und Knotenpunkte etwa 3% des Gesamtstromverbrauchs. Weltweit ist die Kohlendioxid-Produktion durch das Internet vergleichbar mit der des gesamten internationalen Flugverkehrs.*
*(iii) Das Gesamtaufkommen an versendeten Daten im Internet beträgt pro Tag ca. 2 Exabyte beziehungsweise* $2 \cdot 10^9$ *Gigabyte. Die Hälfte davon entfällt auf das Abrufen von Videos.*

## 2. IT-Sicherheit

Unter Sicherheit von Anwendungen in der Informationstechnologie wird der geeignete Umgang mit Daten verstanden, der sicherstellt, dass Risiken wie wirtschaftliche Schäden oder Bedrohungen vermieden werden. Insbesondere sind dabei Datensicherheit, -sicherung und -schutz zu beachten, das heißt die

Vermeidung von Manipulation, Verlust und Verletzung der Vertraulichkeit. Entsprechende Ziele werden mit den Begriffen Vertraulichkeit, Integrität und Verfügbarkeit zusammengefasst:

- Das Ziel der *Vertraulichkeit* stellt sicher, dass zu jedem Zeitpunkt nur autorisierte Personen Zugriff auf die verarbeiteten Daten haben.
- Das Ziel der *Integrität* fordert, dass Änderungen an Daten stets nachvollziehbar sind.
- Das Ziel der *Verfügbarkeit* garantiert, dass auf Daten stets innerhalb eines vorgegebenen Zeitraums zugegriffen werden kann.

Die konkrete Formulierung und Bewertung anwendungsabhängiger Schutzziele erfolgt im Rahmen einer Risikoanalyse. Ein IT-System gilt als sicher, wenn der Aufwand eines Eindringens höher ist als der daraus erzielte Nutzen und die Gefahr von Verlusten durch technische Fehler relativ zum Aufwand der Wiederherstellung der Daten gering ist. Eine absolute Sicherheit ist in den meisten Fällen nicht gerechtfertigt, da dies die Arbeitsfähigkeit stark einschränkt und unangemessen hohe Kosten verursacht. Mögliche Angriffe auf ein System erfolgen meist durch Viren, Identitätsdiebstahl und physischen Einbruch. Als Angriffe gelten auch solche, die durch höhere Gewalt wie Blitzeinschlag verursacht werden. Maßnahmen zur Gewährung der Ziele der IT-Sicherheit sind die räumliche Trennung von Daten, das Einführen von Zugriffskontrollen, die Verwendung von Nutzungsrechten, die regelmäßige Aktualisierung verwendeter Software, die Erstellung von Sicherheitskopien sowie die Verwendung von Antiviren-Software und Firewalls. Gefährdungen der IT-Sicherheit können auch durch Programmierfehler verursacht werden. Gesetzlich untersagt ist jegliche Manipulation fremder Daten sowie das Ausspähen geschützter, das heißt verschlüsselter Daten.

## 3. Datenverschlüsselung

Ein wesentlicher Bestandteil der IT-Sicherheit ist die Verschlüsselung von Daten. Die Entwicklung und Bewertung entsprechender Verfahren wird auch als *Kryptografie* bezeichnet. Entsprechende Ideen existieren seit Jahrtausenden und basierten lange Zeit auf der vertraulichen Vereinbarung eines gemeinsamen geheimen Schlüssels. Dazu war jedes Mal ein vertrauenswürdiger Kurier oder ein persönliches Treffen erforderlich. Vor einigen Jahrzehnten haben sich durch mathematische Theorien und die Verfügbarkeit leistungsfähiger Computer Möglichkeiten ergeben, die diese Schwachstelle vermeiden. Der geheime Schlüssel wird durch offene Kommunikation zwischen Absender und Empfänger mittels sogenannter *Public-Key-Verfahren* generiert. Zentraler Bestandteil dieser Verfahren ist die praktische Irreversibilität gewisser mathematischer Operationen wie dem Multiplizieren von Primzahlen. Wir folgen in diesem Abschnitt den Ausführungen des Buchs *Mathematik sehen und verstehen* von D. Haftendorn (Springer 2016).

**Beispiele 3.1.** *(i) Bei der monoalphabetischen Verschlüsselung werden die Buchstaben des Alphabets zyklisch um eine feste, durch den vereinbarten*

*Schlüssel festgelegte, Anzahl von Positionen verschoben. Mit 25 Tests oder effizienter durch Verwendung von Buchstabenhäufigkeiten können entsprechend verschlüsselte Texte jedoch lesbar gemacht werden. Etwas schwieriger ist dies bei polyalphabetischen Verschlüsselungen, die auf einem Schlüsselwort und einer Buchstabenaustauschtabelle basieren.*
*(ii) Sehr sicher ist das Verschlüsseln eines Texts, wenn dieser zunächst in eine Zahl beziehungsweise Ziffernfolge übersetzt wird, beispielsweise durch Identifikation von Buchstaben mit zweistelligen Zahlen und Hintereinanderfügen dieser Zahlen. Ist $m \in \mathbb{N}$ die zu übertragende Nachricht und $s \in \mathbb{N}$ ein Schlüssel selber Länge, so kann die verschlüsselte Nachricht $c \in \mathbb{N}$ selber Länge definiert werden durch die Ziffern*

$$c_i = (m_i + s_i) \bmod 10.$$

*Der Schlüssel kann beispielsweise als Teilfolge der Zahl $\pi$ gewählt werden. Ohne den Schlüssel ist die Nachricht nicht zu entschlüsseln, da sich jede beliebige Zahl $m'$ selber Länge mit einem geeigneten Schlüssel $s'$ aus $c$ erzeugen lässt.*

Moderne Verfahren der Kryptografie nutzen Methoden der Mathematik, um das Problem der Vereinbarung eines gemeinsamen geheimen Schlüssels zu vermeiden. Wesentlich ist dabei die Verwendung von Primzahlen sowie der Primzahlfaktorisierung beliebiger Zahlen, das heißt die eindeutige Darstellung einer beliebigen Zahl $z \in \mathbb{N}$ als Produkt von Primzahlen $p_1, p_2, \dots, p_k$

$$z = p_1^{\ell_1} p_2^{\ell_2} \dots p_k^{\ell_k}.$$

Zwar lässt sich die bloße Existenz dieser Faktorisierung rigoros nachweisen, jedoch ist die praktische Bestimmung der Faktoren ein NP-schwieriges Problem, das heißt der Aufwand bekannter Verfahren zur Bestimmung der Faktoren wächst exponentiell mit der Anzahl der Stellen von $z$.

**Beispiel 3.2.** *Zur Bestimmung der Primzahlfaktorisierung einer Zahl $z \in \mathbb{N}$ muss jede Primzahl $p \leq \sqrt{z}$ als Faktor getestet werden. Davon gibt es nach einer Formel von Gauß etwa $\sqrt{z}/\ln(\sqrt{z})$ viele. Besitzt $z$ beispielsweise 300 Stellen, so ergeben sich ca. $3 \cdot 10^{147}$ Tests. Selbst bei Einsatz aller verfügbarer Rechner ist dies nur in Milliarden von Jahren realisierbar.*

3.1. **Rechnen modulo $n$.** Zwei ganze Zahlen $x$ und $y$ werden als *gleich modulo $n$* bezeichnet, wenn ihre Differenz ein Vielfaches von $n$ ist, das heißt wenn eine ganze Zahl $q$ existiert, sodass

$$x = y + qn$$

gilt. In diesem Fall schreibt man $x \equiv_n y$; ist $x$ in einer Gleichung $x \equiv_n y$ wählbar, so wählen wir $x$ mit der Eigenschaft $0 \leq x \leq n-1$. Beim Multiplizieren können Faktoren durch gleiche Zahlen modulo $n$ ersetzt werden, das heißt gilt $a \equiv_n \widetilde{a}$, so auch $ab \equiv_n \widetilde{a}b$. Das Potenzieren modulo $n$ ist ein Spezialfall des Multiplizierens.

**Beispiele 3.3.** *(i) Es gilt* $5 \equiv_3 2$ *und* $4 \equiv_4 0$.
*(ii) Es gilt* $5 \cdot 5 \equiv_3 2 \cdot 5 \equiv_3 2 \cdot 2 \equiv_3 1$.
*(iii) Es gilt* $5^4 \equiv_{14} 25 \cdot 25 \equiv_{14} 11 \cdot 11 \equiv_{14} 9$.

Zwei Zahlen $a, b$ heißen *invers modulo* $n$, wenn $ab \equiv_n 1$ gilt.

**Beispiel 3.4.** *Es gilt* $3 \cdot 7 \equiv_{10} 1$.

Ohne Kenntnis der Zahl $n$ ist die Bestimmung eines inversen Elements im Allgemeinen nicht möglich.

**Bemerkungen 3.5.** *(i) Als Folgerung des Eulerschen Satzes ergibt sich der kleine Satz von Fermat, der besagt, dass für jede Primzahl* $p$ *und jede Zahl* $1 \leq a \leq p-1$ *gilt*

$$a^{p-1} \equiv_p 1.$$

*(ii) Zur effizienten Berechnung großer Potenzen modulo* $n$ *eignet sich die sogenannte* Powermod-Methode*. Dabei wird der Exponent in Binärdarstellung geschrieben und eine geeignete Klammerung verwendet, so dass nur wenige Quadrate berechnet werden müssen:*

$$a^{\ell} \equiv_n \left( \left( \ldots \left( a^{b_k} \right)_n^2 \ldots a^{b_2} \right)_n^2 a^{b_1} \right)_n^2 a^{b_0},$$

*sofern* $\ell = b_k 2^k + \ldots b_1 2 + b_0$ *mit* $b_i \in \{0, 1\}$ *und mit der Notation* $(x)_n = x \bmod n$.

Eine Erweiterung des Euklidischen Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen erlaubt für gegebene Zahlen $a, b$ die Bestimmung zweier Zahlen $s, t$, sodass gilt

$$1 = sa + tb$$

und insbesondere $1 \equiv_a tb$, das heißt $t$ ist invers zu $b$ modulo $a$.

3.2. **Kryptografische Verfahren.** Das Diffie–Hellman-Protokoll ist ein symmetrisches Verfahren zur Festlegung eines gemeinsamen geheimen Schlüssels, der mittels offener Kommunikation zwischen den beteiligten Parteien, die im Folgenden mit Anton und Berta bezeichnet werden, festlegt.

**Algorithmus 3.6** (Diffie–Hellman-Protokoll)**.**
*(1) Anton und Berta wählen offen eine Primzahl* $p$ *und eine Zahl* $1 < g < p$.
*(2) Anton wählt eine geheime Zahl* $1 < s < p$*, berechnet* $a \equiv_p g^s$ *und teilt Berta die Zahl* $a$ *mit.*
*(3) Berta wählt eine geheime Zahl* $1 < r < p$*, berechnet* $b \equiv_p g^r$ *und teilt Anton die Zahl* $b$ *mit.*
*(4) Anton berechnet den geheimen Schlüssel* $k \equiv_p b^s$.
*(5) Berta berechnet den geheimen Schlüssel* $k \equiv_p a^r$.

Anton und Berta sind nach Ausführung des Protokolls im Besitz desselben Schlüssels, denn es gilt

$$k_{\text{Anton}} \equiv_p b^s \equiv_p (g^r)^s \equiv_p (g^s)^r \equiv (a)^r \equiv_p k_{\text{Berta}}.$$

Ein Angreifer kann die Zahlen $p, g$ und $a, b$ abhören. Um jedoch an den Schlüssel $k$ zu gelangen, muss er eine der Gleichungen

$$g^s \equiv_p a, \quad g^r \equiv_p b$$

nach $s$ oder $r$ lösen, was für große Zahlen jedoch praktisch unmöglich ist.

**Beispiel 3.7.** *Die Gleichung* $7^s \equiv_{23} 14$ *kann durch Ausprobieren von* $s = 1, 2, \ldots, 22$ *gelöst werden und liefert die Lösung* $s = 15$*. Bei einer Primzahl* $p$ *mit* $\ell$ *Stellen sind jedoch etwa* $10^\ell$ *viele Zahlen zu testen. In der Praxis werden Primzahlen mit 300 Stellen verwendet.*

Die einfach auszuwertende diskrete Exponentialabbildung $s \mapsto g^s \bmod p$ ist nach Sätzen der Algebra über zyklische Gruppen bijektiv, sofern $g$ eine Primitivwurzel der Restklassengruppe modulo $p$ ist. In diesem Fall heißt die Umkehrabbildung diskreter Logarithmus. Seine unregelmäßige Verteilung ist in Abbildung 3 illustriert und veranschaulicht, wieso die Berechnung des diskreten Logarithmus ein nach heutiger Kenntnis NP-schwieriges Problem ist. Dieser Sachverhalt und die Kommutativität der Potenz, das heißt die Identität

$$(g^s)^r = (g^r)^s,$$

sind die Basis des Diffie-Hellman-Protokolls. Eine effizente algorithmische Bestimmung von Primitivwurzeln ist zwar nicht bekannt, jedoch können die offenen Paare $(p, g)$ je nach erforderlicher Schlüssellänge aus Tabellen entnommen werden. Die diskrete Exponentialabbildung ist ein Beispiel einer in der Kryptographie als *Einwegfunktion* bezeichnete Abbildung.

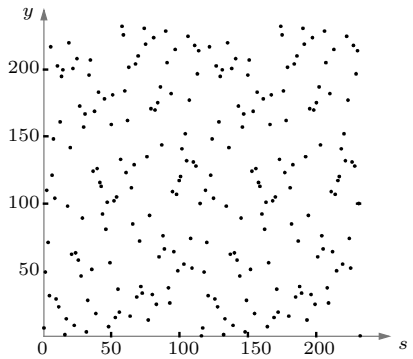


Abbildung 3. Unregelmäßige Verteilung der Werte der diskreten Exponentialfunktion $s \mapsto y = b^s \bmod p$, $1 \le s \le p-1$, mit $p = 233$ und $b = 7$.

Um gewisse Schwächen des Diffie-Hellman-Verfahren zu vermeiden, entwickelten Rivest, Shamir und Adleman das nach ihnen benannte RSA-Verfahren, welchem die praktische Irreversibilität der Primzahlfaktorisierung zugrunde liegt. Hierbei handelt es sich um ein asymmetrisches Verfahren, das heißt die Verschlüsselung erfolgt durch den Versender der Nachricht mit den vom Empfänger öffentlich bereitgestellten Informationen.

**Algorithmus 3.8** (RSA-Protokoll, Schlüsselerzeugung)**.**
*(1) Anton wählt Primzahlen $p, q$ und bestimmt $n = pq$ sowie den Wert $\varphi(n) = (p-1)(q-1)$ der Eulerschen $\varphi$-Funktion.*
*(2) Anton wählt $1 < e < \varphi$ mit $\mathrm{ggT}(e, \varphi) = 1$ und bestimmt eine Zahl $d$ mit $1 \leq d \leq \varphi$ und de $\equiv_\varphi 1$, die geheim gehalten wird. Die Zahlen $p, q, \varphi$ werden gelöscht.*
*(3) Anton veröffentlicht den Schlüssel $(n, e)$.*

Will Berta eine verschlüsselte Nachricht $m$ an Anton schicken, geht sie folgendermaßen vor.

**Algorithmus 3.9** (RSA-Protokoll, Anwendung)**.**
*(1) Berta lädt Antons öffentlichen Schlüssel $(n, e)$.*
*(2) Berta bestimmt $c \equiv_n m^e$ und schickt $c$ an Anton.*
*(3) Anton erhält $c$ und bestimmt $m' \equiv_n c^d$.*

Dass Anton die richtige Nachricht erhält, dass also $m = m'$ gilt, folgt aus dem Satz von Euler beziehungsweise dem kleinen Satz von Fermat: es gilt

$$m^{k\varphi(n)} \equiv_p m^{k(p-1)(q-1)} \equiv_p \left(m^{(p-1)}\right)^{k(q-1)} \equiv_p 1,$$
$$m^{k\varphi(n)} \equiv_q m^{k(p-1)(q-1)} \equiv_q \left(m^{(q-1)}\right)^{k(p-1)} \equiv_q 1,$$

und aus diesen beiden Gleichungen ergibt sich unmittelbar

$$m^{k\varphi(n)} \equiv_{pq} 1,$$

woraus unter Verwendung von $n = pq$ folgt, dass

$$m' \equiv_n c^d \equiv_n (m^e)^d \equiv_n m^{1+k\varphi(n)} \equiv_n m$$

gilt. Ein Angreifer müsste zur Entschlüsselung von $c$ die Zahl $d$ bestimmen, was jedoch ohne Kenntnis von $\varphi(n)$ beziehungsweise $p$ und $q$ nicht möglich ist. Mit dem RSA-Verfahren lassen sich auch digitale Signaturen erzeugen, die zum Zwecke der Datenintegrität verwendet werden können. Wenn Anton eine Nachricht veröffentlicht, fügt er seiner Nachricht die Signatur $\mathrm{sig} \equiv_n m^d$ hinzu. Ein Leser der Nachricht kann dann durch Berechnung von $\mathrm{test} \equiv_n \mathrm{sig}^e$ prüfen, ob tatsächlich $\mathrm{test} \equiv_n m$ gilt und er der Nachricht vertrauen kann. Gelegentlich wird $m$ für diesen Zweck mittels einer Hash-Funktion $H$ komprimiert, das heißt man betrachtet $\mathrm{sig} \equiv_n H(m)^d$ und überprüft, ob $\mathrm{test} \equiv_n \mathrm{sig}^e \equiv_n H(m)$ gilt. Durch Einbeziehung einer dritten Stelle kann zusätzlich eine Zertifizierung eines Schlüssels erfolgen.