# Kryptografie verständlich

Ein Fachbuch für
Studierende und Anwender

von
Christof Paar und Jan Pelzl

www.crypto-textbook.com


# Kapitel 1

## Einführung in die Kryptografie und Datensicherheit

(Version: 1. Dezember 2016)

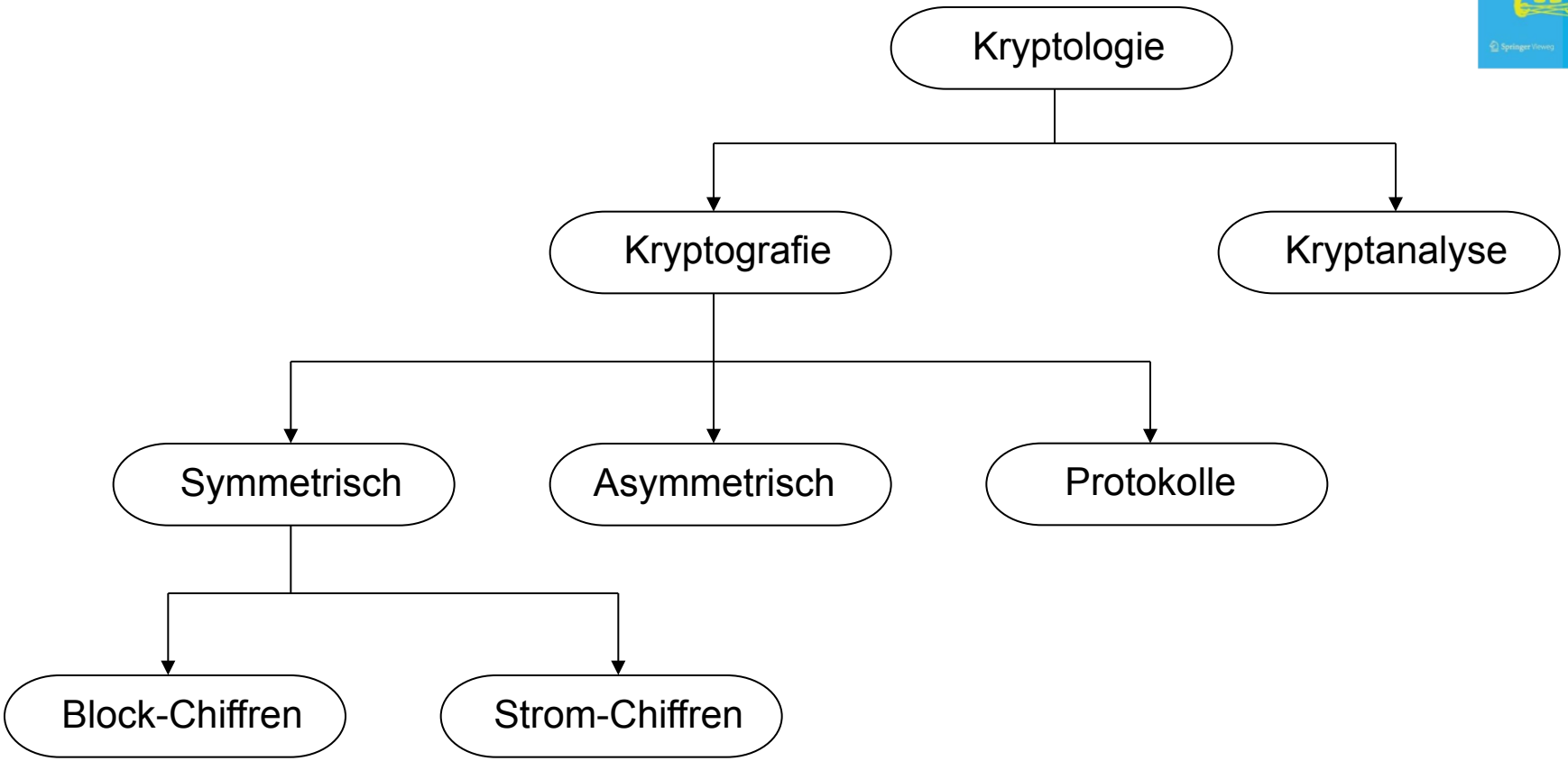
# Übersicht


- Übersicht über das Gebiet der Kryptologie
- Grundlagen der symmetrischen Kryptografie
- Kryptanalyse
- Die Substitutionschiffre
- Modulare Arithmetik
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre

# Übersicht


- **Übersicht über das Gebiet der Kryptologie**
- Grundlagen der symmetrischen Kryptografie
- Kryptanalyse
- Die Substitutionschiffre
- Modulare Arithmetik
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre


# Einführung
## Überblick Kryptologie




## Einführung

Eine kurze Geschichte der Kryptografie

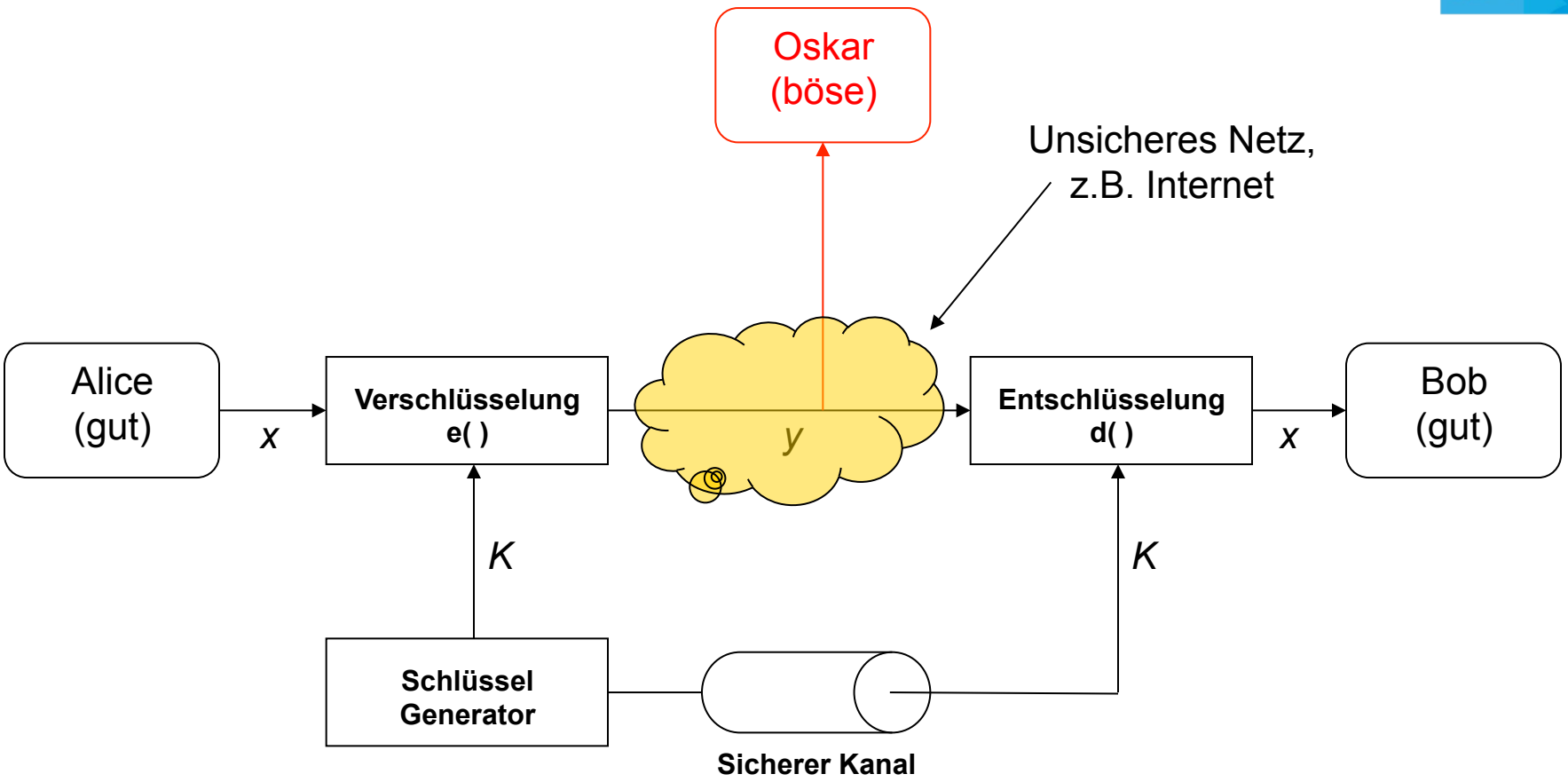
- **Symmetrisch:** Jegliche Ver- und Entschlüsselungsverfahren von der Antike bis 1976.
- **Asymmetrisch:** 1976 wurde das erste asymmetrische Verfahren vorgestellt (Diffie-Hellman Schlüsselaustausch).
- **Hybrider Ansatz:** In heutigen Protokollen verwendet man häufig beide Verfahren, sowohl symmetrische als auch asymmetrische.

# Übersicht


- Übersicht über das Gebiet der Kryptologie
- **Grundlagen der symmetrischen Kryptografie**
- Kryptanalyse
- Die Substitutionschiffre
- Modulare Arithmetik
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre


# Einführung

## Symmetrische Kryptografie: Prinzip




# Einführung
Symmetrische Kryptografie

## Wichtige Begriffe:

- x ist der **Klartext** (engl. plaintext)
- y ist das **Chiffrat** (engl. ciphertext)
- $K$ ist der **Schlüssel** (engl. key)
- $\{K_1, K_2, ..., K_n\}$ ist der **Schlüsselraum** (engl. key space)


# Einführung
## Symmetrische Kryptografie

- Ver- und Entschlüsselung sind inverse Operationen, wenn der selbe Schlüssel K auf beiden Seiten verwendet wird:
  $d_K(y) = d_K(e_K(x)) = x$

| | |
|---|---|
| Verschlüsselung | $\mathbf{y = e_K(x)}$ |
| Entschlüsselung | $\mathbf{x = d_K(y)}$ |

- Wichtig: Der Schlüssel muss auf sicherem Weg zwischen Alice und Bob ausgetauscht werden
- Das System ist nur dann sicher, wenn ein Angreifer den Schlüssel nicht kennt
- ⇒ **Das Problem der sicheren Kommunikation wird auf darauf reduziert, einen Schlüssel sicher zu übertragen und zu speichern**


# Einführung
## Symmetrische Kryptografie

| Schlüssellänge in Bit | Größe des Schlüsselraumes | Klassifizierung der Sicherheit (bei vollständiger Schlüsselsuche) |
|---|---|---|
| 64 | $2^{64}$ | **kurz** (ca. Tage) |
| 80 | $2^{80}$ | **mittel** (ca. 10-20 Jahre) |
| 112/128 | $2^{112}/2^{128}$ | **lang** (mehrere Dekaden), eine mögliche zukünftige Attacke könnten Quantencomputer darstellen |
| 256 | $2^{256}$ | **sehr lang** (mehrere Dekaden) |

# Übersicht


- Übersicht über das Gebiet der Kryptologie
- Grundlagen der symmetrischen Kryptografie
- **Kryptanalyse**
- Die Substitutionschiffre
- Modulare Arithmetik
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre


# Einführung

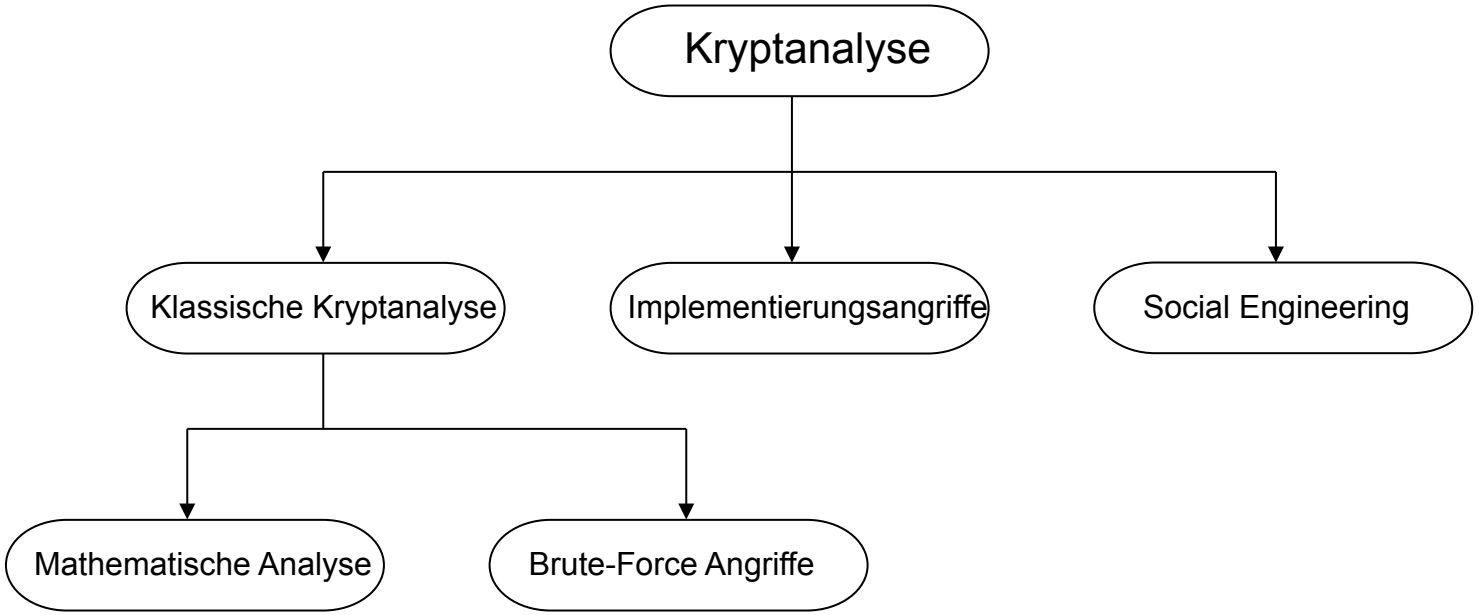
## Kryptanalyse: Definition und Spielregeln

- **Definition**: Die Wissenschaft der Gewinnung des Klartextes aus dem Geheimtext ohne den Schlüssel zu kennen (Oskars Job).

- **Spielregeln**: Die Regeln der Kryptanalyse sind als das **Kerckhoffssche Prinzip** bekannt:
  - Oskar kennt das kryptografische Verfahren (Verschlüsselungs- und Entschlüsselungs-Algorithmen).
  - Oskar kennt nicht den Schlüssel.


# Einführung

## Kryptanalyse: Klassifikation




- **Klassische Angriffe:**
  - Mathematische Analyse
  - Brute-Force Angriffe
- **Implementierungsangriffe**: Gewinnung des Schlüssels durch reverse Engineering oder Seitenkanalmessungen, z.B. für eine Chipkarte
- **Social Engineering**: Z.B. täuschen eines Nutzers, um das Passwort zu bekommen


# Einführung

## Kryptanalyse: Brute-Force Angriff

- Analyse der Chiffre als Black-Box
- Benötigt mindestens ein Paar Klartext/ Chiffrat ($x_0$, $y_0$)
- Teste alle möglichen Schlüssel bis folgende Bedingung erfüllt ist (ausführliche Schlüsselsuche):

$$\mathbf{d_K(y_0) = x_0}$$

- Wie groß sollte der Schlüsselraum sein?


# Einführung
## Exkurs: Passwörter

- Stellen Sie Regeln für gute Passwörter auf

- Z.B.:
  - Zeichenraum ausschöpfen (d.h. bei ASCII-Zeichen auch 256 Möglichkeiten pro Zeichen nutzen)
  - Keine normalen Wörter verwenden (Problem der Wörterbuchattacken)
  - Ausreichende Länge verwenden (mind. 16 Zeichen, je nach Anwendung)
  - Regelmäßig ändern
  - Kein bereits existierendes Passwort verwenden


# Einführung
## Exkurs: Passwörter

- Welche Passwortlänge wählen wir?

- Es kommt darauf an:
  - Abhängig von der zugrundeliegenden Kryptografie, z.B.
    - DES (56 Bit Schlüssellänge) ~7 Zeichen „ausreichend“
    - AES (256 Bit Schlüssellänge) ~32 Zeichen notwendig
  - Abhängig von dem Schutzbedarf und dem möglichen Schaden, sollte das Passwort bekannt werden


# Einführung
## Übung Passwörter

Welche Bitsicherheit hat ein Passwort

1. Mit nur Klein- und Großbuchstaben und Ziffern (0,...,9) ohne Umlaute mit 16 Zeichen Länge?
2. Mit 5 zufällig gewählten ASCII-Zeichen (je 256 Möglichkeiten)?
3. Mit 12 zufällig gewählten ASCII-Zeichen (je 256 Möglichkeiten)?

Lösung:

1. ~2^(95,3)
2. 2^40
3. 2^96 (ungefähr wie 1.)


# Einführung
## Exkurs: Passwörter

- Welche Alternativen zu Passwörtern für die sichere Authentisierung kennen Sie?
  - Chipkarte
  - Token
  - RSA - Token
  - Fingerprint
  - Iris-Scan
  - ...

# Übersicht


- Übersicht über das Gebiet der Kryptologie
- Grundlagen der symmetrischen Kryptografie
- Kryptanalyse
- **Die Substitutionschiffre**
- Modulare Arithmetik
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre


## Einführung

Beispiel: Die Substitutions-Chiffre

**Ziel**: Verschlüsselung eines Textes (im Gegensatz zu Bits)

**Idee**: Ersetze jeden Buchstaben durch einen anderen.
Die Regel der Ersetzung stellt den Schlüssel dar.

**Beispiel:**

- A → K
- B → D
- ....

**Angriffe**: Ist eine **Brute-Force Attacke** (d.h. alle möglichen Schlüssel ausprobieren) möglich?

- Mögliche Schlüssel = 26 x 25 x... 3 x 2 x 1 = 26! ≈ $2^{88}$
- Eine komplette Suche des Schlüsselraums ist mit heutiger Computer-Technologie nicht machbar!


# **Einführung**
## Beispiel: Die Substitutions-Chiffre

## Beispiel für ein Chiffrat

```
VPY YPLNPN GE GQFPN CJYKS UFKSN JUC VLUC?
PG LGN CPY ZFNPY WLN GPLUPW HLUC;
PY SFN CPU HUFDPU VESO LU CPW FYW,
PY IFGGN LSU GLKSPY, PY SFPON LSU VFYW.

WPLU GESU, VFG DLYBGN CJ GE DFUB CPLU BPGLKSN?
GLPSGN ZFNPY, CJ CPU PYOHEPULB ULKSN?
CPU PYOPUHEPULB WLN HYEU JUC GKSVPLI?
WPLU GESU, PG LGN PLU UPDPOGNYPLI.

CJ OLPDPG HLUC, HEWW, BPS WLN WLY!
BFY GKSEPUP GQLPOP GQLPO LKS WLN CLY;
WFUKS DJUNP DOJWPU GLUC FU CPW GNYFUC,
WPLUP WJNNPY SFN WFUKS BJPOCPU BPVFUC.

[...]
```

## Übung: Entschlüsseln Sie das Chiffrat!


## Einführung

Beispiel: Die Substitutions-Chiffre

### Andere Angriffe?

Häufigkeitsanalyse der Buchstaben funktioniert!

- Häufigkeit der Buchstaben eines Klartextes:

  Englisch: „e“ ist der häufigste Buchstabe mit rund 13%, „t“ ⇒ 9%, „a“ ⇒ 8%, ...

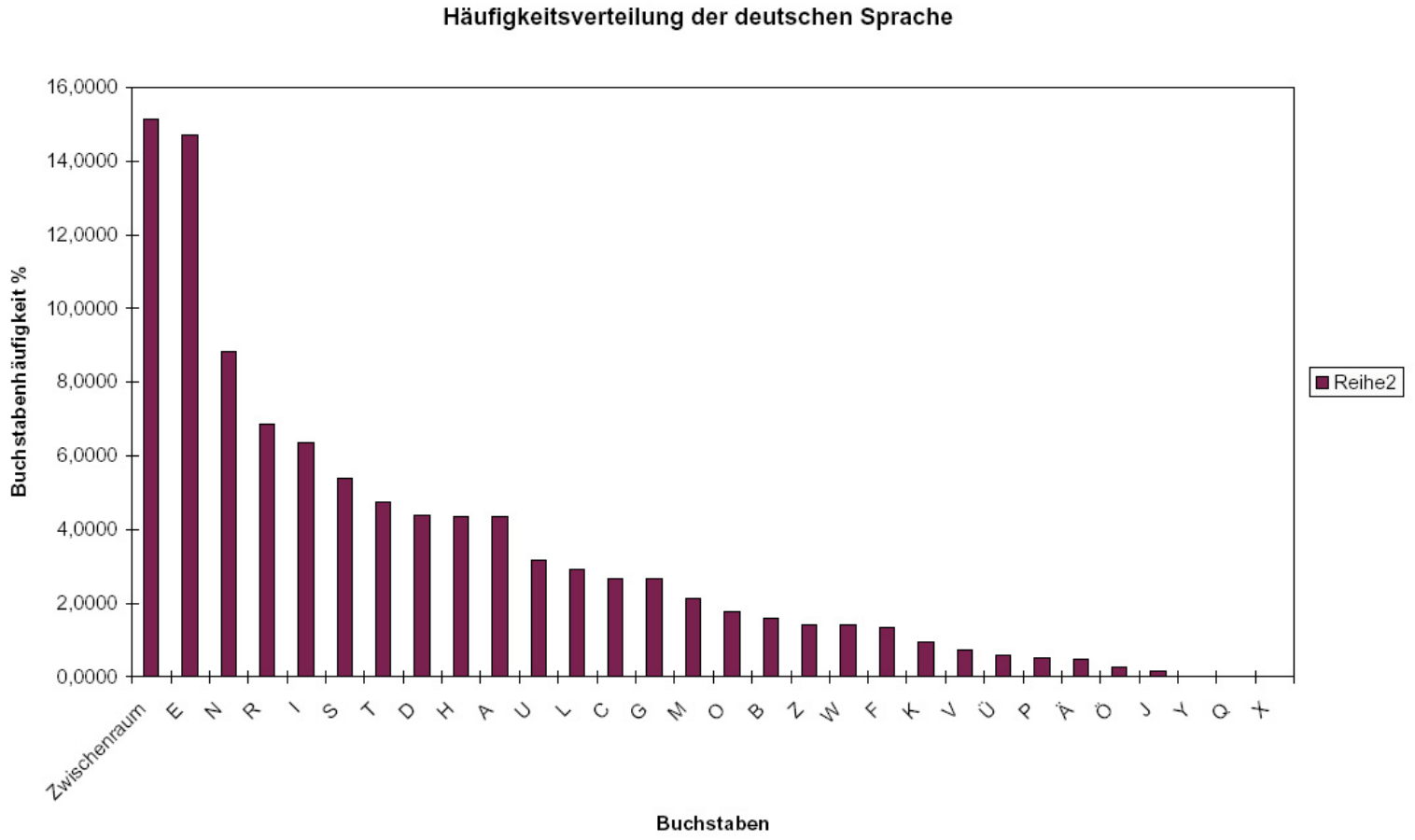
- Häufigkeitsanalyse von Buchstabenpaaren (oder Tripel, Quadrupel, ...).

  Z.B. folgt  auf „q“ fast immer ein „u“...

- Häufigkeit bestimmter Worte

  Englisch: Wörter wie „the“, „and“, ...


# Einführung
## Beispiel: Die Substitutions-Chiffre




# Einführung
## Beispiel: Die Substitutions-Chiffre

Z.B. Häufigster Buchstabe: P

```
VPY YPLNPN GE GQFPN CJYKS UFKSN JUC VLUC?
PG LGN CPY ZFNPY WLN GPLUPW HLUC;
PY SFN CPU HUFDPU VESO LU CPW FYW,
PY IFGGN LSU GLKSPY, PY SFPON LSU VFYW.

WPLU GESU, VFG DLYBGN CJ GE DFUB CPLU BPGLKSN?
GLPSGN ZFNPY, CJ CPU PYOHEPULB ULKSN?
CPU PYOPUHEPULB WLN HYEU JUC GKSVPLI?
WPLU GESU, PG LGN PLU UPDPOGNYPLI.

CJ OLPDPG HLUC, HEWW, BPS WLN WLY!
BFY GKSEPUP GQLPOP GQLPO LKS WLN CLY;
WFUKS DJUNP DOJWPU GLUC FU CPW GNYFUC,
WPLUP WJNNPY SFN WFUKS BJPOCPU BPVFUC.

[...]
```

Vermutung: P → E


# Einführung
## Beispiel: Die Substitutions-Chiffre

Gesuchter Klartext:

```
Wer reitet so spaet durch Nacht und Wind?
Es ist der Vater mit seinem Kind;
Er hat den Knaben wohl in dem Arm,
Er fasst ihn sicher, er haelt ihn warm.

Mein Sohn, was birgst du so bang dein Gesicht?
Siehst Vater, du den Erlkoenig nicht?
Den Erlenkoenig mit Kron und Schweif?
Mein Sohn, es ist ein Nebelstreif.

Du liebes Kind, komm, geh mit mir!
Gar schoene Spiele spiel ich mit dir;
Manch bunte Blumen sind an dem Strand,
Meine Mutter hat manch guelden Gewand.

[...]
```


## Einführung
Beispiel: Die Substitutions-Chiffre

- In der Praxis wurden (werden) die drei zuvor beschriebenen Verfahren verwendet und **kombiniert** um die Substitutions-Chiffre zu brechen.

- Fazit: Gute Verschlüsselungsverfahren müssen die statistische Häufigkeit der zugrunde liegenden Klartexte verbergen. Optimal ist eine **scheinbar zufällige Abfolge der Zeichen eines Chiffrates**.

# Übersicht


- Übersicht über das Gebiet der Kryptologie
- Grundlagen der symmetrischen Kryptografie
- Kryptanalyse
- Die Substitutionschiffre
- **Modulare Arithmetik**
- Verschiebe (oder Caesar) Chiffre und Affine Chiffre


# Einführung

Einführung in die modulare Arithmetik

## Warum benötigen wir modulare Arithmetik?

- Wesentliche Grundlage für asymmetrische Kryptografie (z.B. RSA, DH, elliptische Kurven etc.)
- Einige historische Chiffren können mit modularer Arithmetik elegant beschrieben werden (z.B. die Cäsar- und die affine Chiffre).


# Einführung

## Einführung in die modulare Arithmetik

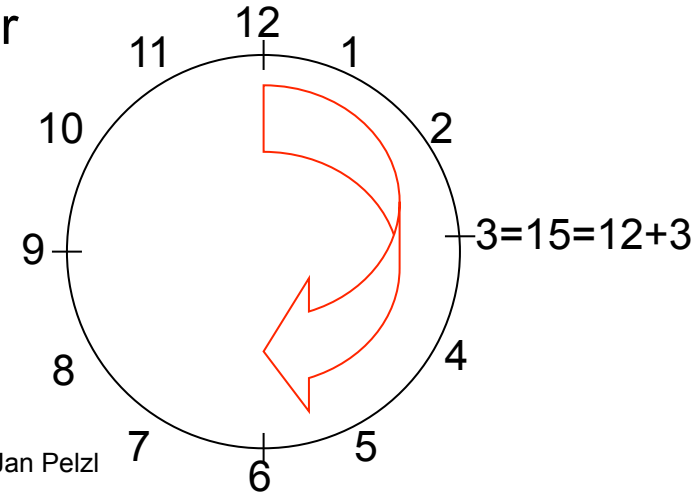
Die meisten Kryptosysteme basieren auf Mengen von Zahlen, die

- **diskret** sind (insbesondere die Menge der ganzen Zahlen)

- **endlich** sind (d.h., wir verwenden nur eine endliche Menge an Zahlen)

Klingt das zu abstrakt?

Schauen wir uns ein bekanntes Beispiel an: Die Stunden auf dem Ziffernblatt einer Uhr




# Einführung

## Einführung in die modulare Arithmetik

- Ziel: Entwicklung eines Systems mit einer endlichen Menge ganzer Zahlen, indem wir **rechnen** können
  (vgl. Beispiel mit den 12 ganzen Zahlen eines Ziffernblatts)

- Benötigt: Eine Operation, die die Zahlen „in Grenzen“ hält (z.B. nicht größer als 12)


# Einführung
## Modulare Arithmetik: Definition Modulo Operation

- **Modulo Operation**

**Definition: Modulo Operation**

Seien *a, r, m* ganze Zahlen und $m > 0$. Dann schreiben wir

$$a \equiv r \bmod m$$

wenn *(r-a) durch m teilbar ist.*

- "*m*" nennt man den **Modul**
- "*r*" nennt man den **Rest**


# Einführung
## Modulare Arithmetik: Beispiele

Beispiele für die modulare Reduktion

- Seien a = 12 und m= 9: $12 \equiv 3 \bmod 9$
- Seien a = 37 und m= 9: $34 \equiv 7 \bmod 9$
- Seien a = -7 und m= 9: $-7 \equiv 2 \bmod 9$

(prüfen Sie, ob die Bedingung „*m* teilt (*r*-*a*)“ in jedem der drei Fälle gilt!)


# Einführung
## Modulare Arithmetik: Eigenschaften (1)

- Der Rest ist nicht eindeutig
- Überraschenderweise gibt es für jeden gegebenen Modul m und eine Zahl a (unendlich) viele gültige Reste
- Beispiel:
  - $12 \equiv 3 \bmod 9 \quad \rightarrow$ 3 ist ein gültiger Rest, da 9 Teiler von (3-12)
  - $12 \equiv 21 \bmod 9 \rightarrow$ 21 ist ein gültiger Rest, da 9 Teiler von (21-12)
  - $12 \equiv -6 \bmod 9 \rightarrow$ -6 ist ein gültiger Rest, da 9 Teiler von (-6-12)


# Einführung
## Modulare Arithmetik: Eigenschaften (2)

- Welchen Rest wählen wir?
- Per Konvention wählen wir die kleinste positive ganze Zahl r als Rest:

Quotient

$$a = q * m + r \quad \text{mit } 0 \le r \le m-1$$

Rest

- Beispiel: a=12 und m= 9

$$12 = 1 * 9 + 3 \quad \rightarrow \quad r = 3$$

- Anmerkung: Dies ist lediglich eine Konvention. Es steht uns frei, jeden beliebigen anderen gültigen Rest für unsere Berechnungen zu verwenden.


# Einführung
## Modulare Arithmetik: Eigenschaften (3)

- Wie funktioniert die modulare Division?
- **Eine Division ist eine Multiplikation mit der Inversen**, d.h.

$$b / a \equiv b * a^{-1} \bmod m$$

- Die Inverse $a^{-1}$ einer Zahl ist definiert als

$$a * a^{-1} \equiv 1 \bmod m$$

- Beispiel: Was ergibt 5 / 7 mod 9?
  - Die Inverse von 7 mod 9 ist 4 da $7 * 4 \equiv 28 \equiv 1 \bmod 9$
  - Daher: $5 / 7 \equiv 5 * 4 = 20 \equiv 2 \bmod 9$


# Einführung

## Modulare Arithmetik: Eigenschaften (4)

- Wie wird die Inverse berechnet?
- Die Inverse a mod m existiert nur, wenn gilt

  ggT(a,m) = 1

  Anmerkung: In obigem Beispiel ist ggT(5, 9) = 1, so dass die Inverse von 5 modulo 9 existiert

  Hinweis: Im Moment kennen wir keinen anderen Weg als Ausprobieren, um die Inverse zu finden. Später werden wir hierfür den erweiterten Euklidischen Algorithmus kennenlernen.


# Einführung

## Modulare Arithmetik: Eigenschaften (5)

- Wir können zu jeder Zeit einer Berechnung eine modulare Reduktion durchführen
- Beispiel einer modularen Exponentiation: $3^8 \bmod 7$
  - Ansatz 1 (Exponentiation gefolgt von einer modularen Reduktion): $3^8 = 6561 \equiv 2 \bmod 7$
    Anmerkung: Zwischenergebnis ist viel größer als 6!
  - Ansatz 2 (Exponentiation mit Reduktion von Zwischenergebnissen):

    $3^8 = 3^4 * 3^4 = 81 * 81$

    Reduktion des Zwischenergebnisses 81:

    $3^8 = 81 * 81 \equiv 4 * 4 \bmod 7$

    $4 * 4 = 16 \equiv 2 \bmod 7$

    Anmerkung: Diese Berechnung ist ohne Taschenrechner einfach möglich!


# Einführung
## Modulare Arithmetik: Eigenschaften (6)

- Allgemeine Regel:

**Bei den meisten Algorithmen ist es vorteilhaft,**

**Zwischenergebnisse so früh wie möglich zu reduzieren.**


# Einführung

## Modulare Arithmetik: Der Ring $Z_m$

- Wir drücken nun modulare Arithmetik als Menge mit definierten Operationen aus und erhalten den Ring der ganzen Zahlen $Z_m$ mit folgenden Eigenschaften

- **Abgeschlossenheit**: Das Ergebnis einer Addition oder Multiplikation ist wieder im Ring
- Addition und Multiplikation sind **assoziativ**, d.h. für alle $a,b,c \in Z_m$

  $a + (b + c) = (a + b) + c$

  $a \times (b \times c) = (a \times b) \times c$

  und die Addition ist **kommutativ**: $a + b = b + a$
- Es gilt das **Distributivgesetz**: $a\times(b+c) = (a\times b)+(a\times c)$ für alle $a,b,c \in Z_m$
- Es gibt ein **neutrales Element *0* bezüglich der Addition**, d.h. für alle $a \in Z_m$

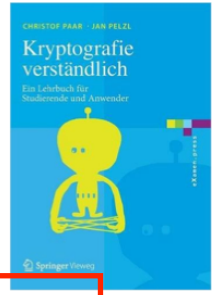
  $a + 0 \equiv a \bmod m$
- Für alle $a \in Z_m$, gibt es immer ein **additives inverses Element *–a*** mit

  $a + (-a) \equiv 0 \bmod m$
- Es gibt ein **neutrale Element 1 bezüglich der Multiplikation**, d.h. für alle $a \in Z_m$

  $a \times 1 \equiv a \bmod m$
- Die **multiplikative Inverse $a^{-1}$**

  $a \times a^{-1} \equiv 1 \bmod m$

  existiert nur für einige, jedoch nicht alle Elemente in $Z_m$.




# Einführung

## Modulare Arithmetik: Der Ring $Z_m$ (2)

Vereinfacht gesagt: Ein Ring ist eine Struktur, in welcher wir immer addieren und multiplizieren können und manchmal dividieren können.

- Ein Element a $\varepsilon$ $Z_m$ hat genau dann eine multiplikative Inverse, wenn ggT(a,m) = 1, d.h. wenn a relativ prim zu m ist

- Beispiel: Der Ring $Z_9$ = {0,1,2,3,4,5,6,7,8}
- Die Elemente 0, 3 und 6 haben keine Inverse, da diese relativ prim zu 9 sind
- Die Inversen der anderen Elemente 1, 2, 4, 5, 7 und 8 sind:
- $1^{-1} \equiv 1 \bmod 9$ $2^{-1} \equiv 5 \bmod 9$ $4^{-1} \equiv 7 \bmod 9$
- $5^{-1} \equiv 2 \bmod 9$ $7^{-1} \equiv 4 \bmod 9$ $8^{-1} \equiv 8 \bmod 9$

# Übersicht


- Übersicht über das Gebiet der Kryptologie
- Grundlagen der symmetrischen Kryptografie
- Kryptanalyse
- Die Substitutionschiffre
- Modulare Arithmetik
- **Verschiebe- (oder Caesar-) Chiffre und Affine Chiffre**


# Einführung
## Verschiebe- oder Cäsar-Chiffre (1)

- Historische Chiffre, u.a. angeblich von Julius Caesar verwendet
- Idee: Ersetze Buchstaben durch andere Buchstaben
- Regel: Buchstaben werden um k Positionen im Alphabet verschoben
- Benötige Abbildung Buchstaben → Zahlen:

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

- Beispiel für $k = 7$
- Klartext = `ATTACK` = 0, 19, 19, 0, 2, 10
- Chiffrat = `haahr` = 7, 0, 0, 7, 17
- Bemerkung: Am Ende des Alphabets beginnen die Buchstaben wieder von vorne, was mathematisch als modulare Reduktion modulo 26 ausgedrückt werden kann, z.B. $19 + 7 = 26 \equiv 0 \bmod 26$


# Einführung
## Verschiebe- oder Cäsar-Chiffre (2)

- Elegante mathematische Beschreibung der Chiffre:

Seien k, x, y ε {0,1, …, 25}

- Verschlüsselung: $y = e_k(x) \equiv x + k \bmod 26$
- Entschlüsselung: $x = d_k(x) \equiv y - k \bmod 26$

- Frage: Ist die Chiffre sicher?
- Antwort: Nein. Es sind zahlreiche Angriffe möglich. Z.B.
  - Ausführliche Schlüsselsuche (es gibt nur 26 Schlüssel!)
  - Häufigkeitsanalyse, vergl. Substitutionschiffre

# Einführung
## Affine Chiffre (1)


- Erweiterung der Chiffre durch zusätzliche Multiplikation mit einem weiteren Schlüssel-Wert
- Schlüssel besteht nun aus zwei Werten: k = (a,b)

Seien k, x, y ε {0,1, …, 25}

- Verschlüsselung: $y = e_k(x) \equiv a\,x + b \bmod 26$
- Entschlüsselung: $x = d_k(x) \equiv a^{-1}(y - b) \bmod 26$

- Da die Inverse von a für die Entschlüsselung benötigt wird, können wir nur Werte für nehmen mit ggT(a,m) = 1
  (Es gibt 12 Werte mit dieser Bedingung.)
- Daraus folgt die Größe des Schlüsselraums #k = 12*26 = 312
- Sicherheit: Es sind wieder zahlreiche Angriff möglich, u.a.
  - Ausführliche Schlüsselsuche
  - Häufigkeitsanalyse

# Lessons Learned (1)


- Eine langer Schlüssel alleine garantiert noch nicht die Sicherheit eines Algorithmus: das Verfahren könnte immer noch mit analytischen Methoden angegriffen werden (siehe Substitutions-Chiffre).

- **Entwickeln Sie NIE einen eigenen kryptografischen Algorithmus**. Wenn Sie dennoch einen Algorithmus entwickeln, dann lassen Sie diesen durch ein Team von erfahrenen Kryptographen überprüfen.

- Benutzen Sie nur **überprüfte und getestete Algorithmen** (symmetrische, asymmetrische, Hash-Funktionen) und Protokolle.

# Lessons Learned (2)


- Zwei wichtige Aspekte der **Langzeit-Sicherheit**:
  - Der Zeitraum, in dem die Implementierung der kryptografischen Algorithmen genutzt werden (meistens ein paar Jahre)
  - Der Zeitraum, in dem die verschlüsselten Daten geheim gehalten werden sollen (abhängig von der Anwendung kann es sich dabei um einige Jahrzehnte handeln)

- **Schlüssellängen** von symmetrischen Algorithmen, die einen Brute-Force Angriff verhindern:
  - **64 Bit - unsicher**, außer für Daten mit sehr kurzem Wert.
  - **112-128 Bit – Langzeit-Sicherheit** für einige Jahrzehnte, höchstwahrscheinlich auch gegen Attacken von Geheimdiensten (außer diese Organisationen würden Quanten-Computer einsetzen, welche zur Zeit nicht existieren und wahrscheinlich auch nie werden).

- Modulare Arithmetik ist eine Grundlage für viele moderne Chiffren und kann dazu verwendet werden, historische Chiffren darzustellen