

# Einführung in die Kryptologie

Johannes Köbler



Institut für Informatik  
Humboldt-Universität zu Berlin

SS 2022

## Lernziele

- Kryptografische Verfahren schaffen Vertrauen in ungeschützten Umgebungen
- Sie ermöglichen sichere Kommunikation über unsichere Kanäle und können verhindern, dass sich ein Kommunikationspartner unfair verhält
- In unsicheren Umgebungen wie dem Internet können sie die aus direkter Interaktion gewohnte Sicherheit herstellen
- Und auch die Interaktion in sicheren Umgebungen wird um Möglichkeiten erweitert, die ohne Kryptografie nicht denkbar wären
- In diesem Modul werden wir uns mit den mathematischen Grundlagen von kryptografischen Verfahren beschäftigen, wobei (symmetrische und asymmetrische) Verschlüsselungsverfahren im Vordergrund stehen
- Im Mastermodul Kryptologie werden wir dann auch kryptografische Verfahren und Protokolle für andere Schutzziele betrachten wie z.B. Hashverfahren und digitale Signaturen sowie Pseudozufallsgeneratoren

# Kryptosysteme

- Kryptosysteme (Verschlüsselungsverfahren) dienen der Geheimhaltung von Nachrichten bzw. Daten
- Hierzu gibt es auch andere Methoden wie z.B.
  - Physikalische Maßnahmen: Tresor etc.
  - Organisatorische Maßnahmen: einsamer Waldspaziergang etc.
  - Steganografische Maßnahmen: unsichtbare Tinte etc.

# Überblick weiterer Schutzziele

Andererseits können durch kryptografische Verfahren weitere **Schutzziele** realisiert werden wie z.B.

- **Vertraulichkeit**
  - Geheimhaltung
  - Anonymität (z.B. Mobiltelefon)
  - Unbeobachtbarkeit (von Transaktionen)
- **Integrität**
  - von Nachrichten und Daten
- **Zurechenbarkeit**
  - Authentikation
  - Unabstreichbarkeit
  - Identifizierung
- **Verfügbarkeit**
  - von Daten
  - von Rechenressourcen
  - von Informationsdienstleistungen

# Kryptologie

In das Umfeld der Kryptologie fallen die folgenden Begriffe

- **Kryptografie:**

Lehre von der Geheimhaltung von Informationen durch Verschlüsselung  
Im weiteren Sinne: Wissenschaft von der Übermittlung, Speicherung und Verarbeitung von Daten in einer von potentiellen Gegnern bedrohten Umgebung

- **Kryptoanalysis:**

Erforschung der Methoden eines unbefugten Angriffs gegen ein Kryptoverfahren

Zweck: Vereitelung der mit seinem Einsatz verfolgten Ziele

- **Kryptoanalyse:**

Analyse eines Kryptoverfahrens zum Zweck der Bewertung seiner kryptografischen Stärken und Schwächen

- **Kryptologie:**

Wissenschaft vom Entwurf, der Anwendung und der Analyse von kryptografischen Verfahren (umfasst Kryptografie und Kryptoanalyse)

# Abgrenzung zwischen Krypto- und Codesystemen

## Codesysteme

- operieren auf semantischen Einheiten
- starre Festlegung, welche Zeichenfolge wie zu ersetzen ist

## Beispiel (Ausschnitt aus einem Codebuch der deutschen Luftwaffe)

---

xve	<b>Bis auf weiteres Wettermeldung gemäß Funkbefehl testen</b>
yde	<b>Frage</b>
sLk	<b>Befehl</b>
fin	<b>beendet</b>
eom	<b>eigene Maschinen</b>

---

## Kryptosysteme

- operieren auf syntaktischen Einheiten
- flexibler Mechanismus durch Schlüsselvereinbarung

# Alphabet

## Definition

- Ein **Alphabet**  $A = \{a_0, \dots, a_{m-1}\}$  ist eine geordnete endliche Menge von **Zeichen**  $a_i$
- Eine Folge  $x = x_1 \dots x_n \in A^n$  heißt **Wort** (der **Länge**  $n$ )
- Die Menge aller Wörter über dem Alphabet  $A$  ist  $A^* = \bigcup_{n \geq 0} A^n$

## Beispiel

- Das **lateinische Alphabet**  $A_{lat}$  enthält die 26 Buchstaben A, ..., Z
- Bei klassischen Verfahren wurde in Klartexten meist auf den Gebrauch von Interpunktions- und Leerzeichen sowie auf Groß- und Klein-schreibung verzichtet
  - ~~ Verringerung der Redundanz im Klartext

# Mathematische Beschreibung eines Kryptosystems

## Definition

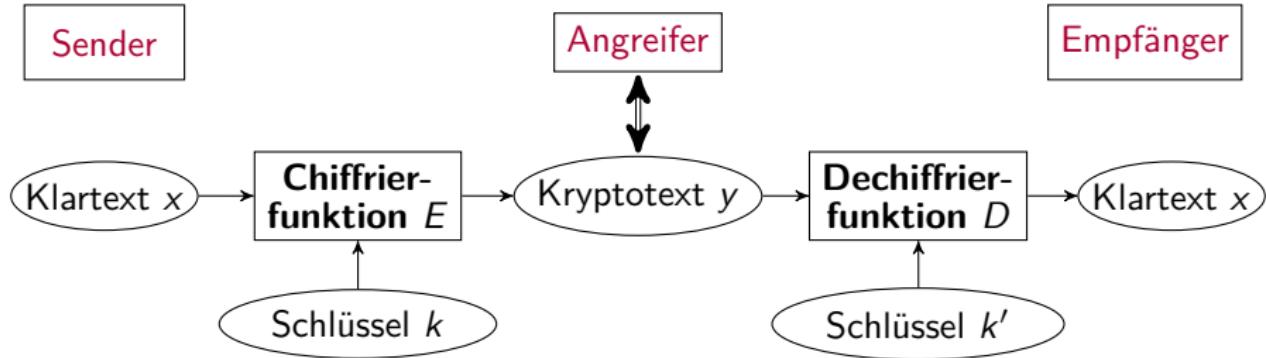
Ein **Kryptosystem** wird durch folgende Komponenten beschrieben:

- $A$ , das **Klartextalphabet**,
- $B$ , das **Kryptotextalphabet**,
- $K$ , der **Schlüsselraum (key space)**,
- $M \subseteq A^*$ , der **Klartextraum (message space)**,
- $C \subseteq B^*$ , der **Kryptotextraum (ciphertext space)**,
- $E : K \times M \rightarrow C$ , die **Verschlüsselungsfunktion (encryption function)**,
- $D : K \times C \rightarrow M$ , die **Entschlüsselungsfunktion (decryption function)**
- $S \subseteq K \times K$ , eine Menge von Schlüsselpaaren  $(k, k')$  mit der Eigenschaft, dass für jeden Klartext  $x \in M$  folgende Beziehung gilt:

$$D(k', E(k, x)) = x \quad (*)$$

Bei symmetrischen Kryptosystemen ist  $S = \{(k, k) \mid k \in K\}$ , weshalb wir in diesem Fall auf die Angabe von  $S$  verzichten können

# Ver- und Entschlüsseln mit einem Kryptosystem



- Zu jedem Schlüssel  $k \in K$  korrespondiert eine
  - **Chiffrierfunktion**  $E_k : x \mapsto E(k, x)$  und eine
  - **Dechiffrierfunktion**  $D_k : y \mapsto D(k, y)$
- Die Gesamtheit dieser Abbildungen wird auch **Chiffre** (englisch **cipher**) genannt
- Daneben wird der Begriff „Chiffre“ auch als Bezeichnung für einzelne Kryptotextzeichen oder kleinere Kryptotextsequenzen verwendet

# Notwendigkeit der Injektivität von Chiffrierfunktionen

## Lemma

Für jedes Paar  $(k, k') \in S$  ist die Chiffrierfunktion  $E_k$  injektiv

## Beweis

- Angenommen, für zwei Klartexte  $x_1$  und  $x_2$  gilt  $E(k, x_1) = E(k, x_2)$
- Dann folgt

$$x_1 \stackrel{(*)}{=} D(k', \underbrace{E(k, x_1)}_{E(k, x_2)}) = D(k', E(k, x_2)) \stackrel{(*)}{=} x_2$$

□

# Moduloarithmetik

Die Moduloarithmetik erlaubt es uns, das Klartextalphabet mit einer Addition und Multiplikation auszustatten

## Definition (teilt-Relation, modulare Kongruenz)

Seien  $a, b, m$  ganze Zahlen mit  $m \geq 1$

- Die Zahl  $a$  teilt  $b$  (kurz:  $a|b$ ), falls ein  $d \in \mathbb{Z}$  existiert mit  $b = ad$
- Teilt  $m$  die Differenz  $a - b$ , so schreiben wir hierfür  $a \equiv_m b$   
(in Worten:  $a$  ist kongruent zu  $b$  modulo  $m$ )
- Weiterhin bezeichne

$$a \bmod m = \min\{a - dm \geq 0 \mid d \in \mathbb{Z}\}$$

den bei der Ganzzahldivision von  $a$  durch  $m$  auftretenden Rest  
(also diejenige ganze Zahl  $r \in \{0, \dots, m-1\}$ , für die eine Zahl  $d \in \mathbb{Z}$  existiert mit  $a = dm + r$ )

- Die Zahl  $d$  wird auch **Ganzzahlquotient** von  $a$  und  $b$  genannt und mit  $a \text{ div } m$  bezeichnet

# Addition und Multiplikation von Buchstaben

- Die auf  $\mathbb{Z}$  definierten Operationen

$$a \oplus_m b := (a + b) \bmod m$$

und

$$a \odot_m b := ab \bmod m$$

sind abgeschlossen auf  $\mathbb{Z}_m = \{0, \dots, m-1\}$  und

- bilden auf dieser Menge einen kommutativen Ring mit Einselement, den sogenannten **Restklassenring modulo  $m$**
- Für  $a \oplus_m -b$  schreiben wir auch  $a \ominus_m b$
- Wenn aus dem Kontext klar ist, dass  $a, b \in \mathbb{Z}_m$  sind, schreiben wir anstelle von  $a \oplus_m b$ ,  $a \ominus_m b$  und  $a \odot_m b$  auch einfach  $a + b$ ,  $a - b$  bzw.  $ab$
- Durch Identifikation der Zeichen  $a_i$  eines Alphabets  $A = \{a_0, \dots, a_{m-1}\}$  mit ihren Indizes können wir die auf  $\mathbb{Z}_m$  definierten Rechenoperationen auf Buchstaben übertragen

# Die additive Chiffre

## Definition (Buchstabenrechnung)

- Sei  $A = \{a_0, \dots, a_{m-1}\}$  ein Alphabet
- Für Indizes  $i, j \in \{0, \dots, m-1\}$  und eine ganze Zahl  $z \in \mathbb{Z}$  definieren wir:

$$\begin{aligned} a_i + a_j &= a_{i \oplus_m j}, & a_i - a_j &= a_{i \ominus_m j}, & a_i a_j &= a_{i \odot_m j}, \\ a_i + z &= a_{i \oplus_m z}, & a_i - z &= a_{i \ominus_m z}, & za_j &= a_{z \odot_m j} \end{aligned}$$

Mit Hilfe dieser Notation lässt sich die Verschiebechiffre, die auch als additive Chiffre bezeichnet wird, leicht beschreiben

## Definition

- Bei der **additiven Chiffre** ist  $A = B = M = C$  ein beliebiges Alphabet mit  $m := |A|$  und  $K = \{0, \dots, m-1\}$
- Für  $k \in K$ ,  $x \in M$  und  $y \in C$  gilt

$$E(k, x) = x + k \text{ und } D(k, y) = y - k$$

# Die ROT13 Verschlüsselung

- Im Fall des lateinischen Alphabets führt der Schlüssel  $k = 13$  auf eine interessante Chiffrierfunktion

---

$x$	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$E(13, x)$	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M

---

- Diese ist in UNIX-Umgebungen unter der Bezeichnung **ROT13** bekannt
- Natürlich kann mit dieser Substitution nicht ernsthaft die Vertraulichkeit von Nachrichten gewahrt werden
- Vielmehr soll durch sie ein unbeabsichtigtes Mitlesen – etwa von Rätsellösungen – verhindert werden
- ROT13 ist eine **involutarische** (also zu sich selbst inverse) Abbildung, d.h. für alle  $x \in A$  gilt  $\text{ROT13}(\text{ROT13}(x)) = x$
- Da ROT13 zudem keinen Buchstaben auf sich selbst abbildet, ist sie sogar **echt involutorisch**

## Die multiplikative Chiffre

- Die Buchstabenrechnung legt folgende Modifikation der Caesar-Chiffre nahe
- Anstatt auf jeden Klartextbuchstaben den Schlüsselwert  $k$  zu addieren, können wir die Klartextbuchstaben auch mit  $k$  multiplizieren
- Allerdings erhalten wir hierbei nicht für jeden Wert von  $k$  eine injektive Chiffrierfunktion
- So bildet etwa die Funktion  $g : A_{lat} \rightarrow A_{lat}$  mit  $g(x) = 2x$  sowohl  $A$  als auch  $\mathbb{N}$  auf den Buchstaben  $g(A) = g(\mathbb{N}) = A$  ab
- Um eine hinreichende und notwendige Bedingung für die Zulässigkeit eines Schlüsselwerts  $k$  formulieren zu können, führen wir zunächst eine Reihe von zahlentheoretischen Begriffen ein

# Begriffe der elementaren Zahlentheorie

**Definition.** Seien  $a, b \in \mathbb{Z}$ .

- Für  $(a, b) \neq (0, 0)$  ist  $\text{ggT}(a, b) = \max\{d \in \mathbb{Z} \mid d|a \wedge d|b\}$  der **größte gemeinsame Teiler** von  $a$  und  $b$
- Für  $a \neq 0, b \neq 0$  ist  $\text{kgV}(a, b) = \min\{d \in \mathbb{Z} \mid d \geq 1 \wedge a|d \wedge b|d\}$  das **kleinste gemeinsame Vielfache** von  $a$  und  $b$
- Ist  $\text{ggT}(a, b) = 1$ , so nennt man  $a$  und  $b$  **teilerfremd** oder man sagt,  $a$  ist **relativ prim** zu  $b$

**Lemma.** Seien  $a, b, c \in \mathbb{Z}$  mit  $(a, b) \neq (0, 0)$ . Dann gilt

- $\text{ggT}(a, b) = \text{ggT}(b, a + bc)$  und somit
- $\text{ggT}(a, b) = \text{ggT}(b, a \bmod b)$ , falls  $b \geq 1$  ist

**Beweis.**

Ein Teiler von  $a$  und  $b$  ist auch Teiler von  $b$  und  $a + bc$  und umgekehrt  $\square$

# Euklidischer Algorithmus

- Der größte gemeinsame Teiler zweier Zahlen  $a$  und  $b$  lässt sich wie folgt bestimmen (o. B. d. A. gelte  $a > b > 0$ )
- Bestimme durch Division mit Rest natürliche Zahlen

$r_0 = a > r_1 = b > r_2 > \dots > r_s > r_{s+1} = 0$  und  $d_1, \dots, d_s$  mit

$$r_{i-1} = d_i r_i + r_{i+1} \text{ für } i = 1, \dots, s$$

(also  $d_i = r_{i-1} \text{ div } r_i$  und  $r_{i+1} = r_{i-1} \text{ mod } r_i$ )

- Hierzu sind  $s$  Divisionsschritte erforderlich
- Wegen

$$\text{ggT}(r_{i-1}, r_i) = \text{ggT}(r_i, \underbrace{r_{i-1} - d_i r_i}_{r_{i+1}})$$

folgt  $\text{ggT}(a, b) = \text{ggT}(r_s, r_{s+1}) = r_s$

# Euklidischer Algorithmus

Der euklidische Alg. kann iterativ oder rekursiv implementiert werden

**Prozedur** Euklid<sub>it</sub>( $a, b$ )

```

1 repeat
2    $r := a \bmod b$ 
3    $a := b$ 
4    $b := r$ 
5 until  $r = 0$ 
6 return( $a$ )

```

**Prozedur** Euklid<sub>rek</sub>( $a, b$ )

```

1 if  $b = 0$  then
2   return( $a$ )
3 else
4   return(Euklidrek( $b, a \bmod b$ ))

```

## Beispiel

Für  $a = 693$  und  $b = 147$  erhalten  
wir  $\text{ggT}(693, 147) = r_4 = 21$

$$\begin{array}{rcccl}
 i & r_{i-1} & = & d_i \cdot & r_i + r_{i+1} \\
 \hline
 1 & 693 & = & 4 \cdot 147 + & 105 \\
 2 & 147 & = & 1 \cdot 105 + & 42 \\
 3 & 105 & = & 2 \cdot 42 + & 21 \\
 4 & 42 & = & 2 \cdot 21 + & 0
 \end{array}$$



# Laufzeit des euklidischen Algorithmus

- Zur Abschätzung von  $s$  verwenden wir die Folge der Fibonacci-Zahlen

$$F_n = \begin{cases} 0, & \text{falls } n = 0 \\ 1, & \text{falls } n = 1 \\ F_{n-1} + F_{n-2}, & \text{falls } n \geq 2 \end{cases}$$

- Induktiv über  $i = s+1, s, \dots, 0$  folgt  $r_i \geq F_{s+1-i}$  und somit  $a = r_0 \geq F_{s+1}$
- Induktiv über  $n \geq 0$  folgt  $F_{n+1} \geq \phi^{n-1}$  für  $\phi = \frac{1+\sqrt{5}}{2}$  (**goldener Schnitt**):

- Der Induktionsanfang ( $n = 0$  oder  $1$ ) ist klar:

$$F_2 = F_1 = 1 = \phi^0 \geq \phi^{-1}$$

- Unter der Induktionsvoraussetzung (IV)  $F_{i+1} \geq \phi^{i-1}$  für  $i \leq n-1$  folgt wegen  $\phi^2 = \phi + 1$

$$F_{n+1} = F_n + F_{n-1} \geq \phi^{n-2} + \phi^{n-3} = \phi^{n-3}(\phi + 1) = \phi^{n-1}$$

- Somit ist  $a \geq \phi^{s-1}$ , d. h.  $s \leq 1 + \lfloor \log_\phi a \rfloor$

# Laufzeit des euklidischen Algorithmus

Als Folgerung erhalten wir folgende Laufzeitabschätzung

## Satz

- Seien  $a > b > 0$  ganze Zahlen und sei  $n$  die Länge von  $a$  in Binärdarstellung
- Dann führt der euklidische Algorithmus  $O(n)$  Divisionsschritte zur Berechnung von  $\text{ggT}(a, b)$  durch
- Dies führt auf eine Zeitkomplexität von  $O(n^3)$ , da jede Ganzzahldivision in Zeit  $O(n^2)$  durchführbar ist

## Erweiterter euklidischer Algorithmus

- Der euklidische Algorithmus kann so modifiziert werden, dass er eine lineare Darstellung des ggT liefert:

$$\text{ggT}(a, b) = \lambda a + \mu b \text{ mit } \lambda, \mu \in \mathbb{Z} \text{ (ebenfalls in Zeit } O(n^3))$$

- Hierzu werden für  $i = 0, \dots, s$  neben  $r_i$  und  $d_i$  weitere Zahlen bestimmt:

$$p_0 = 1, p_1 = 0 \text{ und } p_i = p_{i-2} - d_{i-1}p_{i-1} \text{ für } i = 2, \dots, s$$

$$q_0 = 0, q_1 = 1 \text{ und } q_i = q_{i-2} - d_{i-1}q_{i-1} \text{ für } i = 2, \dots, s$$

- Dann gilt die Gleichung  $ap_i + bq_i = r_i$  für  $i = 0$  und  $i = 1$  und wegen

$$\begin{aligned} ap_{i+1} + bq_{i+1} &= a(p_{i-1} - d_i p_i) + b(q_{i-1} - d_i q_i) \\ &= ap_{i-1} + bq_{i-1} - d_i(ap_i + bq_i) \\ &= (r_{i-1} - d_i r_i) \\ &= r_{i+1} \end{aligned}$$

folgt induktiv über  $i = 2, \dots, s$ , dass sie auch für  $i = s$  gilt:

$$ap_s + bq_s = r_s = \text{ggT}(a, b)$$

# Erweiterter euklidischer Algorithmus

Korollar (Lemma von Bezout)

Der größte gemeinsame Teiler von  $a$  und  $b$  ist in der Form

$$\text{ggT}(a, b) = \lambda a + \mu b \text{ mit } \lambda, \mu \in \mathbb{Z}$$

darstellbar

Beispiel

Für  $g = \text{ggT}(693, 147)$  ergibt sich die Darstellung  $g = 3 \cdot 693 - 14 \cdot 147 = 21$ :

$i$	$r_{i-1} = d_i \cdot r_i + r_{i+1}$	$p_i$	$q_i$	$p_i \cdot 693 + q_i \cdot 147 = r_i$
0		1	0	$1 \cdot 693 + 0 \cdot 147 = 693$
1	$693 = 4 \cdot 147 + 105$	0	1	$0 \cdot 693 + 1 \cdot 147 = 147$
2	$147 = 1 \cdot 105 + 42$	1	-4	$1 \cdot 693 - 4 \cdot 147 = 105$
3	$105 = 2 \cdot 42 + 21$	-1	5	$-1 \cdot 693 + 5 \cdot 147 = 42$
4	$42 = 2 \cdot 21 + 0$	3	-14	$3 \cdot 693 - 14 \cdot 147 = 21$



# Schlussfolgerungen aus dem Lemma von Bezout

## Korollar

Der ggT( $a, b$ ) wird von allen gemeinsamen Teilern von  $a$  und  $b$  geteilt:

$$x|a \wedge x|b \Rightarrow x|\text{ggT}(a, b).$$

## Beweis

Seien  $\mu, \lambda \in \mathbb{Z}$  mit  $\mu a + \lambda b = \text{ggT}(a, b)$ . Falls  $x$  sowohl  $a$  als auch  $b$  teilt, dann teilt  $x$  auch die Produkte  $\mu a$  und  $\lambda b$  und somit auch deren Summe  $\square$

## Korollar

$$\text{ggT}(a, b) = \min\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$$

## Beweis

Sei  $M = \{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$ ,  $m = \min M$  und  $g = \text{ggT}(a, b)$ . Wegen

- $g \in M$  folgt dann  $g \geq m$  und da
- $g$  jede Zahl in  $M$  teilt, folgt auch  $g \leq m$

 $\square$

# Schlussfolgerungen aus dem Lemma von Bezout

## Korollar

$$\text{ggT}(a, b) = \min\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\} \quad (*)$$

## Korollar

$$\text{ggT}(a, m) = \text{ggT}(b, m) = 1 \Leftrightarrow \text{ggT}(ab, m) = 1$$

## Beweis.

- Da  $a$  und  $b$  teilerfremd zu  $m$  sind, existieren Zahlen  $\mu, \lambda, \mu', \lambda' \in \mathbb{Z}$  mit  $\mu a + \lambda m = 1 = \mu' b + \lambda' m$
- Wegen

$$1 = (\mu a + \lambda m)(\mu' b + \lambda' m) = \underbrace{\mu \mu' ab}_{\mu''} + \underbrace{(\mu a \lambda' + \mu' b \lambda + \lambda' m)}_{\lambda''} m$$

folgt dann  $\text{ggT}(ab, m) = 1$  aus Gleichung  $(*)$

- Gilt umgekehrt  $\text{ggT}(ab, m) = 1$ , so existieren Zahlen  $\mu, \lambda \in \mathbb{Z}$  mit  $\mu ab + \lambda m = 1$ . Mit  $(*)$  folgt nun sofort  $\text{ggT}(a, m) = \text{ggT}(b, m) = 1$   $\square$

# Schlussfolgerungen aus dem Lemma von Bezout

## Korollar (Lemma von Euklid)

Sind  $a$  und  $b$  teilerfremd und teilt  $a$  das Produkt  $bc$ , so teilt  $a$  auch  $c$ :

$$\text{ggT}(a, b) = 1 \wedge a|bc \Rightarrow a|c$$

## Beweis

- Wegen  $\text{ggT}(a, b) = 1$  existieren Zahlen  $\mu, \lambda \in \mathbb{Z}$  mit  $\mu a + \lambda b = 1$
- Falls  $a$  das Produkt  $bc$  teilt, muss  $a$  auch die Zahl  $\mu ac + \lambda bc = c$  teilen

□

## Zulässige Schlüssel für die multiplikative Chiffre

- Sei  $A$  ein Alphabet und sei  $g : A \rightarrow A$  eine Abbildung der Form  $g(x) = bx$
- Damit  $g$  injektiv (oder gleichbedeutend, surjektiv) ist, muss es zu jedem Zeichen  $y \in A$  genau ein Zeichen  $x \in A$  mit  $bx = y$  geben
- Der folgende Satz gibt hierfür eine notwendige und hinreichende Bedingung an

Satz. Seien  $b, y, m$  ganze Zahlen mit  $m \geq 1$ .

Dann besitzt die Kongruenz  $bx \equiv_m y$  genau im Fall  $\text{ggT}(b, m) = 1$  eine eindeutige Lösung  $x \in \{0, \dots, m - 1\}$

# Zulässige Schlüssel für die multiplikative Chiffre

Satz. Seien  $b, y, m$  ganze Zahlen mit  $m \geq 1$ .

Dann besitzt die Kongruenz  $bx \equiv_m y$  genau im Fall  $\text{ggT}(b, m) = 1$  eine eindeutige Lösung  $x \in \{0, \dots, m - 1\}$

## Beweis.

- Im Fall  $\text{ggT}(b, m) = g > 1$  ist wegen  $b(m/g) = (b/g)m \equiv_m 0$  mit  $x$  auch  $x' = x + m/g \not\equiv_m x$  eine Lösung von  $bx \equiv_m y$  und somit die Kongruenz  $bx \equiv_m y$  nicht eindeutig lösbar
- Umgekehrt folgt im Fall  $\text{ggT}(b, m) = 1$  aus den beiden Kongruenzen  
 $bx_1 \equiv_m y$  und  $bx_2 \equiv_m y$   
 sofort  $b(x_1 - x_2) \equiv_m 0$ , also  $m|b(x_1 - x_2)$
- Mit dem Lemma von Euklid folgt  $m|(x_1 - x_2)$ , d.h.  $x_1 \equiv_m x_2$
- Folglich ist die Abbildung  $f : \mathbb{Z}_m \rightarrow \mathbb{Z}_m$  mit  $f(x) = bx \bmod m$  injektiv und somit auch surjektiv (da Definitions- und Wertebereich übereinstimmen), d.h. die Kongruenz  $bx \equiv_m y$  hat genau eine Lösung in  $\mathbb{Z}_m$   $\square$

# Zulässige Schlüssel für die multiplikative Chiffre

## Korollar

Im Fall  $\text{ggT}(b, m) = 1$  hat die Kongruenz  $bx \equiv_m 1$  genau eine Lösung, die das **multiplikative Inverse von  $b$  modulo  $m$**  genannt und mit  $b^{-1} \bmod m$  (oder einfach mit  $b^{-1}$ ) bezeichnet wird

- Wegen  $\text{ggT}(a, m) = \text{ggT}(b, m) = 1 \Rightarrow \text{ggT}(ab, m) = 1$  ist die Menge

$$\mathbb{Z}_m^* = \{b \in \mathbb{Z}_m \mid \text{ggT}(b, m) = 1\}$$

aller invertierbaren Elemente von  $\mathbb{Z}_m$  unter der Operation  $\odot_m$  abgeschlossen, d.h.  $(\mathbb{Z}_m^*, \odot_m, 1)$  bildet eine multiplikative Gruppe

- Allgemeiner zeigt man, dass die Multiplikation eines beliebigen Rings  $(R, +, \cdot, 0, 1)$  mit Eins auf der Menge

$$R^* = \{a \in R \mid \exists b \in R : ab = 1 = ba\}$$

aller **Einheiten von  $R$**  eine Gruppe bildet (siehe Übungen)

- Diese Gruppe  $(R^*, \cdot, 1)$  wird als **Einheitengruppe von  $R$**  bezeichnet

## Berechnung von multiplikativen Inversen

- Das multiplikative Inverse von  $b$  modulo  $m$  ergibt sich aus der linearen Darstellung

$$\lambda b + \mu m = \text{ggT}(b, m) = 1$$

zu  $b^{-1} = \lambda \bmod m$

- Die folgende Tabelle gibt für jedes  $b \in \mathbb{Z}_{26}^*$  das multiplikative Inverse an:

$b$	1	3	5	7	9	11	15	17	19	21	23	25
$b^{-1}$	1	9	21	15	3	19	7	23	11	5	17	25

- Bei Kenntnis von  $b^{-1}$  kann die Kongruenz  $bx \equiv_m y$  leicht zu  $x = yb^{-1} \bmod m$  gelöst werden

# Die affine Chiffre

## Definition

Bei der **affinen Chiffre** ist  $A = B = M = C$  ein beliebiges Alphabet mit  $m := |A|$  und  $K = \mathbb{Z}_m^* \times \mathbb{Z}_m$ . Für  $k = (b, c) \in K$ ,  $x \in M$  und  $y \in C$  gilt

$$E(k, x) = bx + c \quad \text{und} \quad D(k, y) = b^{-1}(y - c)$$

- Hierbei liefert die Schlüsselkomponente  $b = -1$  für jedes  $c \in \mathbb{Z}_m$  eine involutorische Chiffrierfunktion  $E_{(-1,c)}(x) = c - x$
- Wählen wir für  $c$  ebenfalls den Wert  $-1$ , so ergibt sich die Funktion  $E_{(-1,-1)} = -x - 1$ , die auch als **revertiertes Alphabet** bekannt ist:

---

$x$	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$-x$	A Z Y X W V U T S R Q P O N M L K J I H G F E D C B
$-x - 1$	Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

---

- Offenbar ist  $E_{(-1,-1)}$  genau dann echt involutorisch, wenn  $m$  gerade ist

# Die affine Chiffre

## Beispiel (affine Chiffre)

- Sei  $A = \{A, \dots, Z\} = B$ , also  $m = 26$ , und sei  $k = (b, c) = (9, 2)$
- Um den Klartextbuchstaben  $x = F$  zu verschlüsseln, berechnen wir

$$E(k, x) = bx + c = 9F + 2 = V,$$

da  $F$  den Index 5 und  $V$  den Index 21 hat und  $9 \cdot 5 + 2 = 47 \equiv_{26} 21$  ist

- Für die Entschlüsselung benötigen wir das Inverse  $b^{-1} = q_3 = 3$ :

$$\begin{array}{rclll} i & r_{i-1} & = & d_i \cdot r_i + r_{i+1} & p_i \cdot 26 + q_i \cdot 9 = r_i \\ \hline 0 & & & & 1 \cdot 26 + 0 \cdot 9 = 26 \\ 1 & 26 & = & 2 \cdot 9 + 8 & 0 \cdot 26 + 1 \cdot 9 = 9 \\ 2 & 9 & = & 1 \cdot 8 + 1 & 1 \cdot 26 - 2 \cdot 9 = 8 \\ 3 & 8 & = & 8 \cdot 1 + 0 & -1 \cdot 26 + 3 \cdot 9 = 1 \end{array}$$

- Damit erhalten wir den ursprünglichen Klartextbuchstaben zurück:

$$D(k, y) = b^{-1}(y - c) = 3(V - 2) = F, \text{ da } 3 \cdot 19 = 57 \equiv_{26} 5 \text{ ist} \quad \diamond$$

# Die Eulersche Phi-Funktion

- Zur Berechnung der Schlüsselzahl der multiplikativen und affinen Chiffre benötigen wir die **Eulersche Phi-Funktion**  $\varphi(m) = |\mathbb{Z}_m^*|$
- Für primes  $m$  ist  $\mathbb{Z}_m^* = \{1, \dots, m-1\}$  (d.h.  $\mathbb{Z}_m^* = [m-1]$ , wobei wir die Menge  $\{1, \dots, n\}$  mit **[n]** bezeichnen)

$m$	1	2	3	4	5	6	7	8	9	10
$\mathbb{Z}_m^*$	{0}	[1]	[2]	{1, 3}	[4]	{1, 5}	[6]	{1, 3, 5, 7}	{1, 2, 4, 5, 7, 8}	{1, 3, 7, 9}
$\varphi(m)$	1	1	2	2	4	2	6	4	6	4

- Wegen  $\mathbb{Z}_{p^k} - \mathbb{Z}_{p^k}^* = \{0, p, 2p, \dots, (p^{k-1}-1)p\}$  folgt zudem
 
$$\varphi(p^k) = p^k - p^{k-1} = p^{k-1}(p-1) \text{ für } k \geq 1$$
- Um hieraus eine Formel für  $\varphi(m)$  zu erhalten, genügt es,  $\varphi(n\ell)$  im Fall  $\text{ggT}(n, \ell) = 1$  in Abhängigkeit von  $\varphi(n)$  und  $\varphi(\ell)$  zu bestimmen
- Hierzu betrachten wir die Abbildung  $f : \mathbb{Z}_{n\ell} \rightarrow \mathbb{Z}_n \times \mathbb{Z}_\ell$  mit

$$f(x) = (x \bmod n, x \bmod \ell)$$

# Eine Berechnungsformel für die Eulersche Phi-Funktion

## Beispiel

- Für  $n = 5$  und  $\ell = 6$  erhalten wir die Funktion  $f : \mathbb{Z}_{30} \rightarrow \mathbb{Z}_5 \times \mathbb{Z}_6$  mit

$x$	0	1	2	3	4	5	6	7	8	9
$f(x)$	(0, 0)	<b>(1, 1)</b>	<b>(2, 2)</b>	<b>(3, 3)</b>	<b>(4, 4)</b>	(0, 5)	<b>(1, 0)</b>	<b>(2, 1)</b>	<b>(3, 2)</b>	<b>(4, 3)</b>

$x$	10	<b>11</b>	12	<b>13</b>	14	15	16	<b>17</b>	18	<b>19</b>
$f(x)$	(0, 4)	<b>(1, 5)</b>	(2, 0)	<b>(3, 1)</b>	<b>(4, 2)</b>	(0, 3)	<b>(1, 4)</b>	<b>(2, 5)</b>	(3, 0)	<b>(4, 1)</b>

$x$	20	21	22	<b>23</b>	24	25	26	27	28	<b>29</b>
$f(x)$	(0, 2)	<b>(1, 3)</b>	(2, 4)	<b>(3, 5)</b>	<b>(4, 0)</b>	(0, 1)	<b>(1, 2)</b>	(2, 3)	<b>(3, 4)</b>	<b>(4, 5)</b>

- Man beachte, dass  $f$  eine Bijektion zwischen  $\mathbb{Z}_{30}$  und  $\mathbb{Z}_5 \times \mathbb{Z}_6$  ist
- Zudem liegt  $f(x) = (y, z)$  genau dann in  $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$ , wenn  $x \in \mathbb{Z}_{30}^*$  ist (die Werte  $x \in \mathbb{Z}_{30}^*$ ,  $y \in \mathbb{Z}_5^*$  und  $z \in \mathbb{Z}_6^*$  sind **fett** gedruckt)
- Folglich bildet  $f$  die  $x$ -Werte in  $\mathbb{Z}_{30}^*$  bijektiv auf die Werte in  $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$  ab

## Beispiel (Schluss)

- Für die Umkehrfunktion  $f^{-1} : \mathbb{Z}_5 \times \mathbb{Z}_6 \rightarrow \mathbb{Z}_{30}$  erhalten wir nebenstehende Tabelle
- Die fett gedruckten Einträge bilden dann die Tabelle der Einschränkung  $\hat{f}^{-1}$  von  $f^{-1}$  auf die Menge  $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$
- Das Bild dieser Einschränkung ist genau die Menge  $\mathbb{Z}_{30}^*$

$f^{-1}$	0	1	2	3	4	5
0	0	25	20	15	10	5
<b>1</b>	6	<b>1</b>	26	21	16	<b>11</b>
2	12	<b>7</b>	2	27	22	<b>17</b>
3	18	<b>13</b>	8	3	28	<b>23</b>
4	24	<b>19</b>	14	9	4	<b>29</b>



- Der chinesische Restsatz (den wir in Kürze beweisen) besagt, dass die Funktion  $f : \mathbb{Z}_{n\ell} \rightarrow \mathbb{Z}_n \times \mathbb{Z}_\ell$  mit  $f(x) = (x \bmod n, x \bmod \ell)$  im Fall  $\text{ggT}(n, \ell) = 1$  bijektiv ist
- Wegen

$$\begin{aligned}\text{ggT}(x, n\ell) = 1 &\Leftrightarrow \text{ggT}(x, n) = \text{ggT}(x, \ell) = 1 \\ &\Leftrightarrow \text{ggT}(x \bmod n, n) = \text{ggT}(x \bmod \ell, \ell) = 1\end{aligned}$$

ist daher die Einschränkung  $\hat{f}$  von  $f$  auf den Bereich  $\mathbb{Z}_{n\ell}^*$  eine Bijektion zwischen  $\mathbb{Z}_{n\ell}^*$  und  $\mathbb{Z}_n^* \times \mathbb{Z}_\ell^*$ , d.h. es gilt

$$\varphi(n\ell) = |\mathbb{Z}_{n\ell}^*| = |\mathbb{Z}_n^* \times \mathbb{Z}_\ell^*| = |\mathbb{Z}_n^*| \cdot |\mathbb{Z}_\ell^*| = \varphi(n)\varphi(\ell)$$

# Eine Berechnungsformel für die Eulersche Phi-Funktion

Als Folgerung aus dem chinesischen Restsatz erhalten wir nun leicht eine Berechnungsformel für die Phi-Funktion

## Satz

Die Eulersche  $\varphi$ -Funktion ist multiplikativ, d. h. für teilerfremde Zahlen  $n$  und  $\ell$  gilt  $\varphi(n\ell) = \varphi(n)\varphi(\ell)$

## Korollar

Sei  $m = \prod_{i=1}^{\ell} p_i^{k_i}$  die Primfaktorzerlegung von  $m$ . Dann gilt

$$\varphi(m) = \prod_{i=1}^{\ell} p_i^{k_i-1}(p_i - 1) = m \prod_{i=1}^{\ell} (p_i - 1)/p_i$$

## Beweis.

Es gilt

$$\varphi\left(\prod_{i=1}^{\ell} p_i^{k_i}\right) = \prod_{i=1}^{\ell} \varphi(p_i^{k_i}) = \prod_{i=1}^{\ell} (p_i^{k_i} - p_i^{k_i-1}) = \prod_{i=1}^{\ell} p_i^{k_i-1}(p_i - 1) \quad \square$$

# Der chinesische Restsatz

- Jede der beiden linearen Kongruenzen

$$x \equiv_3 0$$

$$x \equiv_6 1$$

ist lösbar

- Es gibt aber kein  $x$ , das beide Kongruenzen erfüllt
- Der nächste Satz zeigt, dass unter bestimmten Voraussetzungen gemeinsame Lösungen existieren, und wie sie berechnet werden können

## Satz (Chinesischer Restsatz)

Falls  $m_1, \dots, m_k$  paarweise teilerfremd sind, dann hat das System

$$\begin{aligned} x &\equiv_{m_1} b_1 \\ &\vdots & (*) \\ x &\equiv_{m_k} b_k \end{aligned}$$

für beliebige Zahlen  $b_1, \dots, b_k \in \mathbb{Z}$  genau eine Lösung modulo  $m = \prod_{i=1}^k m_i$

# Der chinesische Restsatz

## Satz (Chinesischer Restsatz)

Falls  $m_1, \dots, m_k$  paarweise teilerfremd sind, dann hat das System

$$\begin{array}{rcl} x & \equiv_{m_1} & b_1 \\ & \vdots & (*) \\ x & \equiv_{m_k} & b_k \end{array}$$

für beliebige Zahlen  $b_1, \dots, b_k \in \mathbb{Z}$  genau eine Lösung modulo  $m = \prod_{i=1}^k m_i$

## Beweis.

- Zu jeder Zahl  $n_i = m/m_i$  ex. wegen  $\text{ggT}(n_i, m_i) = 1$  Zahlen  $\mu_i$  und  $\lambda_i$  mit

$$\mu_i n_i + \lambda_i m_i = \text{ggT}(n_i, m_i) = 1$$

- Für  $i = 1, \dots, k$  löst daher die Zahl  $s_i = \mu_i n_i$  das System

$$x \equiv_{m_j} \begin{cases} 0, & j \neq i \quad (1) \\ 1, & j = i \quad (2) \end{cases}$$

## Beweis des chinesischen Restsatzes

- Zu jeder Zahl  $n_i = m/m_i$  ex. wegen  $\text{ggT}(n_i, m_i) = 1$  Zahlen  $\mu_i$  und  $\lambda_i$  mit

$$\mu_i n_i + \lambda_i m_i = \text{ggT}(n_i, m_i) = 1$$

- Für  $i = 1, \dots, k$  löst daher die Zahl  $s_i = \mu_i n_i$  das System

$$x \equiv_{m_j} \begin{cases} 0, & j \neq i \\ 1, & j = i \end{cases} \quad (1)$$

- Folglich erfüllt  $s = \sum_{i=1}^k b_i s_i$  für  $j = 1, \dots, k$  die Kongruenz

$$s \stackrel{(1)}{\equiv}_{m_j} b_j s_j \stackrel{(2)}{\equiv}_{m_j} b_j,$$

d.h.  $s$  löst das System  $(*)$

- Dies zeigt, dass die Funktion

$$f : \mathbb{Z}_m \rightarrow \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k} \text{ mit } f(x) = (x \bmod m_1, \dots, x \bmod m_k)$$

surjektiv ist

- Da der Definitions- und der Wertebereich von  $f$  gleich groß sind, muss  $f$  auch injektiv sein und  $(*)$  ist eindeutig lösbar

□

## Der chinesische Restsatz

- Man beachte, dass der Beweis des chinesischen Restsatzes konstruktiv ist und die Lösung  $x$  unter Verwendung des erweiterten euklidischen Algorithmus' effizient berechnet werden kann
- Man verifiziert auch leicht, dass  $f$  ein Isomorphismus zwischen dem Restklassenring  $(\mathbb{Z}_m, \oplus_m, \odot_m)$  und dem direkten Produkt der Ringe  $(\mathbb{Z}_{m_i}, \oplus_{m_i}, \odot_{m_i})$ ,  $1 \leq i \leq k$ , ist
- Dies ist nicht nur für theoretische Überlegungen nützlich, sondern hat auch praktische Konsequenzen
- Beispielsweise lässt sich so die Laufzeit von bestimmten Berechnungen im Ring  $\mathbb{Z}_m$  deutlich reduzieren, sofern die Primzahlzerlegung von  $m$  bekannt ist

# Die Hill-Chiffre

- Die von Hill im Jahr 1929 publizierte Chiffre ist eine Erweiterung der multiplikativen Chiffre auf Buchstabenblöcke
- Der Klartext wird also nicht zeichen- sondern blockweise verarbeitet
- Die Blöcke haben eine feste Länge  $\ell$  und sowohl Klar- als auch Kryptotextrum bestehen aus allen Wörtern  $x \in A^\ell$
- Als Schlüssel dient eine  $(\ell \times \ell)$ -Matrix  $k = (k_{ij})$  mit Koeffizienten in  $\mathbb{Z}_m$
- Diese transformiert einen Klartext  $x = x_1 \dots x_\ell \in A^\ell$  in den Kryptotext  $y = y_1 \dots y_\ell$  mit  $y_i = x_1 k_{1i} + \dots + x_\ell k_{\ell i}$  für  $i = 1, \dots, \ell$ :

$$(y_1 \ \dots \ y_\ell) = (x_1 \ \dots \ x_\ell) \begin{pmatrix} k_{11} & \dots & k_{1\ell} \\ \vdots & \ddots & \vdots \\ k_{\ell 1} & \dots & k_{\ell\ell} \end{pmatrix}$$

- Wir bezeichnen die Menge aller  $(\ell \times \ell)$ -Matrizen  $(k_{ij})$  mit Koeffizienten  $k_{ij} \in \mathbb{Z}_m$  mit  $\mathbb{Z}_m^{\ell \times \ell}$

# Die Hill-Chiffre

- Damit der Chiffriervorgang injektiv ist, muss  $k$  invertierbar sein
- Dies lässt sich an der Determinante von  $k$  erkennen

**Definition.** Sei  $R$  ein kommutativer Ring mit 1 und sei  $n \geq 1$

Eine Funktion  $f : R^{n \times n} \rightarrow R$  heißt **Determinantenfunktion**, falls sie für jede Matrix  $A = (a_{ij}) \in R^{n \times n}$  mit Zeilen  $a_1, \dots, a_n \in R^n$  folgende Eigenschaften erfüllt:

- $f$  ist **multilinear**, d.h. für jeden Vektor  $b \in R^n$  und jedes Element  $r \in R$  gilt

$$f \begin{pmatrix} a_1 \\ \vdots \\ r \cdot a_i + b \\ \vdots \\ a_n \end{pmatrix} = r \cdot f \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix} + f \begin{pmatrix} a_1 \\ \vdots \\ b \\ \vdots \\ a_n \end{pmatrix}$$

- $f$  ist **alternierend**, d.h. im Fall  $a_i = a_j$  für  $i \neq j$  gilt  $f(A) = 0$
- $f$  ist **normiert**, d.h.  $f(E_n) = 1$ , wobei  $E_n \in R^{n \times n}$  die Einheitsmatrix ist

# Die Hill-Chiffre

- Tatsächlich ist die Funktion  $f$  durch diese drei Eigenschaften eindeutig festgelegt und wir bezeichnen  $f(A)$  wie üblich mit  $\det(A)$
- Eine explizite Darstellung für die Determinantenfunktion liefert der **Laplacesche Entwicklungssatz**
- Für  $1 \leq i, j \leq n$  sei  $A_{ij}$  die Matrix, die aus  $A$  durch Streichen der  $i$ -ten Zeile und  $j$ -ten Spalte entsteht, und  $\tilde{a}_{ij} = (-1)^{i+j} \det(A_{ij})$  der zugehörige **Kofaktor**
- Dann ist  $\det(A) = a_{11}$  für  $n = 1$  und für  $n \geq 2$  und  $i, j \in [n]$  beliebig ist

$$\det(A) = \sum_{k=1}^n a_{ik} \tilde{a}_{ik} = \sum_{k=1}^n a_{kj} \tilde{a}_{kj}, \quad \begin{array}{l} \text{"Entwicklung nach der } i\text{-ten Zeile} \\ \text{bzw. } j\text{-ten Spalte"} \end{array}$$

- Aus dieser Formel lässt sich zwar ein Algorithmus zur Berechnung der Determinante ableiten, allerdings hat dieser eine exponentielle Laufzeit
- Mit dem **Gaußschen Eliminationsverfahren** lässt sich die Determinante jedoch effizient berechnen (siehe Übungen)

# Die Hill-Chiffre

- Für die Dechiffrierung eines mit der Schlüsselmatrix  $k$  berechneten Kryptotextes wird die inverse Matrix  $k^{-1}$  benötigt
- Invertierbare Matrizen werden auch als **regulär** bezeichnet
- Eine Matrix  $k \in \mathbb{Z}_m^{\ell \times \ell}$  ist genau dann regulär, wenn  $\text{ggT}(\det(k), m) = 1$  ist (siehe Übungen)
- In diesem Fall lässt sich  $k^{-1}$  mit dem Gauß-Jordan-Algorithmus effizient berechnen (siehe Übungen)

## Definition (Hill-Chiffre)

- Sei  $A = \{a_0, \dots, a_{m-1}\}$  ein beliebiges Alphabet und für eine natürliche Zahl  $\ell \geq 2$  sei  $M = C = A^\ell$
- Bei der **Hill-Chiffre** ist  $K = \{k \in \mathbb{Z}_m^{\ell \times \ell} \mid \text{ggT}(\det(k), m) = 1\}$  und es gilt  
 $E(k, x) = xk$  und  $D(k, y) = yk^{-1}$

# Die Hill-Chiffre

## Beispiel

- Wir benutzen zur Chiffrierung von Klartextblöcken der Länge  $\ell = 4$  über dem lateinischen Alphabet  $A_{lat}$  die Schlüsselmatrix

$$k = \begin{pmatrix} 11 & 13 & 8 & 21 \\ 24 & 17 & 3 & 25 \\ 18 & 12 & 23 & 17 \\ 6 & 15 & 2 & 15 \end{pmatrix}$$

- Diese überführt den Klartext HILL in den Kryptotext  $E_k(\text{HILL}) = \text{NERX}$ :

$$(\text{HILL}) \begin{pmatrix} 11 & 13 & 8 & 21 \\ 24 & 17 & 3 & 25 \\ 18 & 12 & 23 & 17 \\ 6 & 15 & 2 & 15 \end{pmatrix} = (\text{NERX}) \quad \text{bzw.} \quad \begin{aligned} 11\text{H} + 24\text{I} + 18\text{L} + 6\text{L} &= N \\ 13\text{H} + 17\text{I} + 12\text{L} + 15\text{L} &= E \\ 8\text{H} + 3\text{I} + 23\text{L} + 2\text{L} &= R \\ 21\text{H} + 25\text{I} + 17\text{L} + 15\text{L} &= X \end{aligned}$$

- Für die Entschlüsselung wird die inverse Matrix  $k^{-1}$  benötigt, die wir in den Übungen berechnen

◀

# Die Vigenère-Chiffre

- Die Vigenère-Chiffre ersetzt den Klartext zeichenweise, allerdings je nach Position im Klartext unterschiedlich
- Sie ist nach dem Franzosen Blaise de Vigenère (1523–1596) benannt

## Definition (Vigenère-Chiffre)

- Sei  $A = B$  ein beliebiges Alphabet
- Die **Vigenère-Chiffre** chiffriert unter einem Schlüssel  $k = k_0 \dots k_{d-1}$  in  $K = A^*$  einen Klartext  $x = x_0 \dots x_{n-1}$  beliebiger Länge zu
  - $E(k, x) = y_0 \dots y_{n-1}$  mit  $y_i = x_i + k_i \bmod d$  für  $i = 0, \dots, n-1$  und dechiffriert einen Kryptotext  $y = y_0 \dots y_{n-1}$  zu
  - $D(k, y) = x_0 \dots x_{n-1}$  mit  $x_i = y_i - k_i \bmod d$  für  $i = 0, \dots, n-1$

# Die Vigenère-Chiffre

## Beispiel

Das Schlüsselwort  $k = WIE$  überführt den Klartext VIGENERE beispielsweise in den Kryptotext

$$\begin{aligned} E(WIE, \text{VIGENERE}) &= \underbrace{V+W}_R \underbrace{I+I}_Q \underbrace{G+E}_K \underbrace{E+W}_A \underbrace{N+I}_V \underbrace{E+E}_I \underbrace{R+W}_N \underbrace{E+I}_M \\ &= RQKAVINNM \end{aligned}$$



- Um einen Klartext  $x$  zu verschlüsseln, wird also das Schlüsselwort  $k = k_0 \dots k_{d-1}$  so oft wiederholt, bis der dabei entstehende **Schlüsselstrom**  $\hat{k} = k_0 k_1 \dots k_{d-1} k_0 k_1 \dots k_{d-1} k_0 k_1 \dots$  die Länge von  $x$  erreicht
- Dann werden  $x$  und  $\hat{k}$  zeichenweise addiert, um den zugehörigen Kryptotext  $y$  zu bilden
- Aus diesem kann der ursprüngliche Klartext  $x$  zurückgewonnen werden, indem man den Schlüsselstrom  $\hat{k}$  wieder subtrahiert

# Die Vigenère-Chiffre

## Beispiel

### Chiffrierung:

$$\begin{array}{rcl} \text{VIGENERE} & (\text{Klartext } x) \\ + \underline{\text{WIEWIEWI}} & (\text{Schlüsselstrom } \hat{k}) \\ \hline \text{RQKAVINM} & (\text{Kryptotext } y) \end{array}$$

### Dechiffrierung:

$$\begin{array}{rcl} \text{RQKAVINM} & (\text{Kryptotext } y) \\ - \underline{\text{WIEWIEWI}} & (\text{Schlüsselstrom } \hat{k}) \\ \hline \text{VIGENERE} & (\text{Klartext } x) \end{array} \quad \triangleleft$$

Die Chiffrierarbeit lässt sich durch Benutzung einer Additionstabelle (auch **Vigenère-Tableau** genannt) erleichtern:

+	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
:	:																									
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

# Die Beaufort-Chiffre

- Um eine involutorische Chiffre zu erhalten, schlug Sir Francis Beaufort, ein Admiral der britischen Marine, vor, den Schlüsselstrom nicht auf den Klartext zu addieren, sondern letzteren von ersterem zu subtrahieren

## Beispiel

- Verschlüsseln wir den Klartext BEAUFORT beispielsweise unter dem Schlüsselwort  $k = WIE$ , so erhalten wir den Kryptotext VEECDQFP

### Chiffrierung:

$$\begin{array}{r} \underline{WIEWIEWI} \quad (\text{Schlüsselstrom}) \\ - \quad \underline{\text{BEAUFORT}} \quad (\text{Klartext}) \\ \hline \underline{\text{VEECDQFP}} \quad (\text{Kryptotext}) \end{array}$$

### Dechiffrierung:

$$\begin{array}{r} \underline{WIEWIEWI} \quad (\text{Schlüsselstrom}) \\ - \quad \underline{\text{VEECDQFP}} \quad (\text{Kryptotext}) \\ \hline \underline{\text{BEAUFORT}} \quad (\text{Klartext}) \end{array}$$

- Eine erneute Verschlüsselung liefert wieder den Klartext BEAUFORT



# Die Autokey-Chiffre

- Die bisher betrachteten Chiffren erzeugen einen **periodischen Schlüsselstrom**  $\hat{k} = \hat{k}_0 \dots \hat{k}_{n-1}$ , das heißt, es gilt  $\hat{k}_i = \hat{k}_{i+d}$  für eine feste Zahl  $d$
- Da dadurch Angriffe erleichtert werden, sollte entweder eine sehr große Periode oder besser ein **fortlaufender Schlüsselstrom** benutzt werden
- Eine Möglichkeit besteht darin, an das Schlüsselwort den Klartext oder den Kryptotext anzuhängen (sogenannte **Autokey-Chiffrierung**)

## Beispiel

- Die Autokey-Chiffre erzeugt mit dem Schlüsselwort **WIE** aus dem Klartext **VIGENERE** folgende Kryptotexte:

Klartext-Schlüsselstrom:

$$\begin{array}{l} \text{VIGENERE } (\text{Klartext}) \\ + \underline{\text{WIEVIGEN}} \quad (\text{Schlüsselstrom}) \\ \hline \text{RQKZVKVR } (\text{Kryptotext}) \end{array}$$

Kryptotext-Schlüsselstrom:

$$\begin{array}{l} \text{VIGENERE } (\text{Klartext}) \\ + \underline{\text{WIERQKVD}} \quad (\text{Schlüsselstrom}) \\ \hline \text{RQKVDOMH } (\text{Kryptotext}) \end{array}$$

# Die Autokey-Chiffre

## Beispiel (Fortsetzung)

- Die Autokey-Chiffre erzeugt mit dem Schlüsselwort *WIE* aus dem Klartext *VIGENERE* folgende Kryptotexte:

Klartext-Schlüsselstrom:

$$\begin{array}{rcl} \text{VIGENERE} & (\text{Klartext}) \\ + \underline{\text{WIEVIGEN}} & (\text{Schlüsselstrom}) \\ \hline \text{RQKZVKVR} & (\text{Kryptotext}) \end{array}$$

Kryptotext-Schlüsselstrom:

$$\begin{array}{rcl} \text{VIGENERE} & (\text{Klartext}) \\ + \underline{\text{WIERQKVD}} & (\text{Schlüsselstrom}) \\ \hline \text{RQKVDOMH} & (\text{Kryptotext}) \end{array}$$

- Auch die Dechiffrierung ist in beiden Fällen einfach:
- Im ersten Fall kann der Empfänger durch Subtraktion des Schlüsselworts den Anfang des Klartextes bilden und so den Schlüsselstrom verlängern
- Noch einfacher ist die Dechiffrierung im zweiten Fall, da sich hier der Schlüsselstrom vom Kryptotext nur durch das vorangestellte Schlüsselwort unterscheidet



# Der One-Time-Pad

- Man kann auch einen zuvor vereinbarten Text aus einem Buch als aperiodischen Schlüsselstrom verwenden (Lauftextverschlüsselung)
- Besser ist es jedoch, mit einem Pseudozufallsgenerator aus einem relativ kurzen Schlüssel einen deutlich längeren Schlüsselstrom zu erzeugen
- Noch besser ist es, den Schlüsselstrom wirklich zufällig zu erzeugen
- Dies führt auf eine absolut sichere Verschlüsselung, sofern der Schlüsselstrom nicht mehrmals benutzt wird
- Ein solcher „Wegwerfschlüssel“ (engl. **One-Time-Pad** oder kurz **OTP**; im Deutschen auch als **individueller Schlüssel** bezeichnet) lässt sich für längere Klartexte allerdings nur mit großem Aufwand generieren und auf einem sicheren Kanal zwischen Sender und Empfänger verteilen
- Der OTP wurde beispielsweise beim „**heißen Draht**“, der 1963 eingerichteten, direkten Fernschreibverbindung zwischen dem Weißen Haus in Washington und dem Kreml in Moskau, angewandt

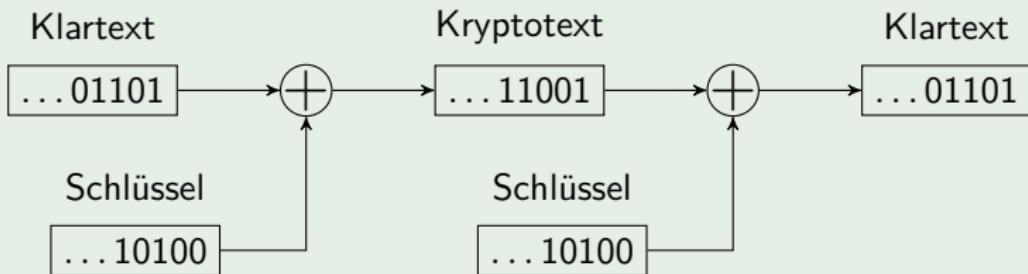
# Der One-Time-Pad

## Beispiel (One-Time-Pad)

- Sei  $A = \{a_0, \dots, a_{m-1}\}$  ein beliebiges Klartextalphabet
- Um einen Klartext  $x = x_0 \dots x_{n-1}$  zu verschlüsseln, wird auf jedes Klartextzeichen  $x_i$  ein neuer, zufällig generierter Schlüsselbuchstabe  $k_i$  addiert:

$$y = y_0 \dots y_{n-1}, \text{ wobei } y_i = x_i + k_i \text{ ist}$$

- Der Klartext wird also wie bei einer additiven Chiffre verschlüsselt, nur dass der Schlüssel nach einmaligem Gebrauch gewechselt wird
- Wie diese ist der One-Time-Pad im Binärfall involutorisch:



# Klassifikation von Kryptosystemen

- Bei den bisher betrachteten Chiffren handelt es sich um **Substitutionen**
- Diese **ersetzen** Klartextzeichen – einzeln oder blockweise – durch Kryptotextsegmente
- Dagegen verändern **Transpositionen** lediglich die *Reihenfolge* der einzelnen Klartextzeichen

## Beispiel (Skytale-Chiffre)

- Die älteste bekannte Verschlüsselungstechnik stammt aus der Antike und wurde im 5. Jahrhundert v. Chr. von den Spartanern entwickelt
- Der Sender wickelt einen Papierstreifen spiralförmig um einen Holzstab (die sog. **Skytale**) und beschreibt ihn mit der Geheimbotschaft:



ÜBERAUS GEHEIMNISVOLL ...  
 ↵ ÜGI...BES...EHV...REO...AIL...UML...SN...

- Der Empfänger benötigt einen Stab mit dem gleichen Umfang, um den Kryptotext auf dem Papierstreifen wieder zu entziffern



# Skytale und Spaltentransposition

- Als Schlüssel fungiert hier also der Stabumfang bzw. die Anzahl  $k$  der Zeilen, mit denen der Stab beschrieben wird
- Ist die Länge  $n = km$  des Klartextes  $x$  ein Vielfaches von  $k$ , so gilt

$$E(k, x_1 \cdots x_{km}) =$$

$$x_1 x_{m+1} \cdots x_{(k-1)m+1} \quad x_2 x_{m+2} \cdots x_{(k-1)m+2} \cdots \quad x_m x_{2m} \cdots x_{km}$$

- Dasselbe Resultat ergibt sich, wenn man  $x$  zeilenweise in eine  $(k \times m)$ -Matrix schreibt und spaltenweise ausliest (sog. **Spaltentransposition**):

---

$x_1$	$x_2$	$\cdots$	$x_m$
$x_{m+1}$	$x_{m+2}$	$\cdots$	$x_{2m}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x_{(k-1)m+1}$	$x_{(k-1)m+2}$	$\cdots$	$x_{km}$

---

# Skytale und Spaltentransposition

- Ist die Länge von  $x$  kein Vielfaches von  $k$ , so kann man  $x$  durch Ein- bzw. Anfügen von sog. **Blenden** (Füllzeichen) verlängern
- Damit der Empfänger diese Füllzeichen nach der Entschlüsselung wieder entfernen kann, ist lediglich darauf zu achten, dass sie nach der Entschlüsselung als solche erkennbar sind
- Von der Methode, die letzte **Zeile** nur teilweise zu beschriften, ist dagegen abzuraten
- In diesem Fall würden nämlich auf dem abgewickelten Papierstreifen Lücken entstehen, die Rückschlüsse auf den benutzten Schlüssel erlauben würden
- Sicherer ist es, wenn der Sender die letzte **Spalte** auf der Skytale nur teilweise beschriftet

# Die Zick-Zack-Transposition

- Eng verwandt mit der Skytale-Chiffre ist die Zick-Zack-Transposition

## Beispiel

Bei Ausführung einer **Zick-Zack-Transposition** wird der Klartext in eine Zick-Zack-Linie geschrieben und horizontal wieder ausgelesen:

Z	Z	L	E		
I	K	A	K	I	I
C	C	N			

ZICKZACKLINIE ~> ZZLEIKAKIICCN

- Als Schlüssel dient also die Höhe der Zick-Zack-Linie
- Dabei werden Zeichen im vorderen Klartextbereich bis fast ans Ende des Kryptotextes verschoben und umgekehrt
- Ein Nachteil hiervon ist, dass bei der Erzeugung des Kryptotextes der gesamte Klartext gepuffert werden muss
- Daher werden meist **Blocktranspositionen** verwendet, die Klartextzeichen nur innerhalb fester Blockgrenzen transponieren

# Die Blocktransposition

## Definition

- Sei  $A = B$  ein Alphabet und für eine Zahl  $\ell \geq 2$  sei  $M = C = A^\ell$
- Eine **Blocktransposition** ordnet jedem Schlüssel  $k \in K$  eine Permutation  $\pi \in S_\ell$  zu, so dass für alle  $x_1 \cdots x_\ell \in M$  und  $y_1 \cdots y_\ell \in C$  gilt:

$$E(k, x_1 \cdots x_\ell) = x_{\pi(1)} \cdots x_{\pi(\ell)} \text{ und } D(k, y_1 \cdots y_\ell) = y_{\pi^{-1}(1)} \cdots y_{\pi^{-1}(\ell)}$$

## Beispiel

Eine Skytale, die mit 4 Zeilen der Länge 6 beschrieben wird, realisiert beispielsweise folgende Blocktransposition:

$i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$\pi(i)$	1	7	13	19	2	8	14	20	3	9	15	21	4	10	16	22	5	11	17	23	6	12	18	24



# Die Blocktransposition

- Für die Entschlüsselung wird die **inverse Permutation**  $\pi^{-1}$  benutzt
- Diese lässt sich leicht berechnen, wenn  $\pi$  durch eine Folge von Zyklen  $(i_1 \ i_2 \ i_3 \ \dots \ i_n)$  dargestellt wird; dies bedeutet, dass  $i_1$  auf  $i_2$ ,  $i_2$  auf  $i_3$ ,  $\dots$  und schließlich  $i_n$  auf  $i_1$  abgebildet wird
- Dabei werden Einerzyklen meist weggelassen, die Zyklen nach der Größe ihrer kleinsten Elemente sortiert und letztere an den Anfang gesetzt

## Beispiel

$i$	1	2	3	4	5	6
$\pi(i)$	4	6	1	3	5	2

$i$	1	2	3	4	5	6
$\pi^{-1}(i)$	3	6	4	1	5	2

- Obiges  $\pi$  hat beispielsweise die Zykeldarstellung  
 $\pi = (1 \ 4 \ 3) \ (2 \ 6) \ (5)$  oder  $\pi = (1 \ 4 \ 3) \ (2 \ 6)$
- Daraus erhalten wir unmittelbar die inverse Permutation  
 $\pi^{-1} = (3 \ 4 \ 1) \ (6 \ 2)$  oder  $\pi^{-1} = (1 \ 3 \ 4) \ (2 \ 6)$



# Die Matrix-Transposition

## Beispiel

- Bei der **Matrix-Transposition** wird der Klartext zeilenweise in eine  $k \times \ell$ -Matrix eingelesen und der Kryptotext spaltenweise gemäß einer Spaltenpermutation  $\pi \in S_\ell$ , die als Schlüssel dient, wieder ausgelesen
- Für  $\pi = (1\ 4\ 3)\ (2\ 6)$  wird also zuerst Spalte  $\pi(1) = 4$ , dann Spalte  $\pi(2) = 6$  und zuletzt Spalte  $\pi(6) = 2$  ausgelesen:

3	6	4	1	5	2
D	I	E	S	E	R
K	L	A	R	T	E
X	T	I	S	T	N
I	C	H	T	S	E
H	R	L	A	N	G

DIESER KLARTEXT IST NICHT SEHR LANG

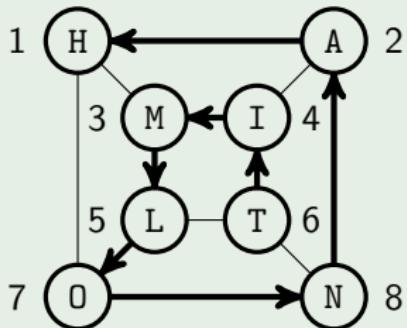
~~> SRSTA RENEG DKXIH EAIHL ETTSN ILTCR



# Die Weg-Transposition

## Beispiel

- Bei der **Weg-Transposition** wird als Schlüssel eine Hamiltonlinie in einem Graphen mit den Knoten  $1, \dots, \ell$  benutzt
- Der Klartext  $x_1 \dots x_\ell$  wird gemäß der Knotennummerierung eingelesen und der Kryptotext entlang der Hamiltonlinie wieder ausgelesen:



HAMILTON  $\rightsquigarrow$  TIMLONAH

- Es ist leicht zu sehen, dass sich jede Blocktransposition durch eine Hamiltonlinie in einem geeigneten Graphen realisieren lässt
- Der Vorteil ist, dass man sich den Verlauf einer Hamiltonlinie in einem Graphen besser einprägen kann als eine Zahlenfolge



# Schlüsselphrasen

- Sehr beliebt ist auch die Methode, sich eine Permutation in Form eines **Schlüsselworts** (oder einer **Schlüsselphrase**) zu merken
- Die zugehörige Permutation  $\sigma$  erhält man, indem man das Wort auf Papier schreibt und in der Zeile darunter die Zeichen abzählt:

Schlüsselwort für $\sigma$	$C \ A \ E \ S \ A \ R$
$i$	1 2 3 4 5 6
$\sigma(i)$	3 1 4 6 2 5
Zyklendarstellung von $\sigma$	(1 3 4 6 5 2)

DIE BLOCKLAENGE IST SECHS  $\rightsquigarrow$   
*EDBOIL LCANKE IGSSET EXCSYH*

- In der nächsten Zeile werden dann die Zeichen gemäß ihrer alphabetischen Reihenfolge abgezählt
- Auf diese Weise erhält man die Wertetabelle von  $\sigma$
- Mehrfach vorkommende Zeichen können entweder gestrichen oder gemäß ihrer Position im Schlüsselwort hochgezählt werden
- Im ersten Fall erhält man eine entsprechend kleinere Blocklänge; in obigem Beispiel wäre dann  $\ell = 5$  und  $\sigma = (1\ 2)(4\ 5)$

# Klassifikation von Substitutionen

- Ein wichtiges Unterscheidungsmerkmal ist die Länge der Klartext- und Kryptotexteinheiten, mit denen eine Substitution operiert

## Definition

- Eine Substitution, die Einzelzeichen ersetzt, heißt **monografisch**, andernfalls **polygrafisch**
- Eine Substitution, die Klartextsegmente durch Einzelzeichen ersetzt, heißt **monopartit**, andernfalls **multipartit**
- Operiert eine Substitution auf Zeichenpaaren, heißt sie **digrafisch**
- Wird der Kryptotext aus Zeichenpaaren gebildet, heißt sie **bipartit**

# Die Porta-Chiffre

- Das älteste bekannte polygrafische Chiffrierverfahren wurde von Giovanni Porta im Jahr 1563 veröffentlicht
- Dabei werden je zwei aufeinanderfolgende Klartextzeichen durch ein einzelnes Kryptotextzeichen ersetzt (d.h. die Chiffre ist monopartit)

## Beispiel

- Bei der **Porta-Chiffre** werden 400 (!) unterschiedliche von Porta für diesen Zweck entworfene Kryptotextzeichen verwendet
- Diese sind in einer  $(20 \times 20)$ -Matrix  $M = (y_{ij})$  angeordnet, deren Zeilen und Spalten mit den 20 Buchstaben A, ..., I, L, ..., T, V, Z indiziert sind
- Zur Ersetzung eines Zeichenpaares  $a_i a_j$  im Klartext wird das in Zeile  $i$  und Spalte  $j$  befindliche Kryptotextzeichen  $E(M, a_i a_j) = y_{ij}$  benutzt ◀

# Die Polybios-Chiffre

- Ein frühes (monografisches) Beispiel einer bipartiten Chiffriermethode geht auf Polybios (circa 200–120 v. Chr.) zurück:

	0	1	2	3	4
0	A	B	C	D	E
1	F	G	H	I	J
2	K	L	M	N	O
3	P	Q	R	S	T
4	U	V	W	X/Y	Z

POLYBIOS  
 ↵ 30 24 21 43 01 13 24 33

- Die **Polybios-Chiffre** benutzt als Schlüssel eine  $5 \times 5$ -Matrix, deren Einträge aus den Klartextzeichen bestehen
- Jedes Zeichen des Klartextes wird durch sein Koordinatenpaar  $ij$  in der Matrix ersetzt
- Der Kryptotextrraum besteht also aus den Ziffernpaaren  $\{00, 01, \dots, 44\}$

# Klassifikation von Substitutionen

- Ob bei der Ersetzung der Klartextsegmente eine fixe oder variable Strategie verfolgt wird, führt auf ein weiteres Unterscheidungsmerkmal

## Definition

Ersetzt eine Substitution Segmente unabhängig von ihrer Position im Klartext, so heißt sie **monoalphabetisch**, sonst **polyalphabetisch**

- Die Ersetzungsregel einer polyalphabetischen Substitution variiert also in Abhängigkeit von den bereits verarbeiteten Klartextsegmenten
- Die Bezeichnung „monoalphabetisch“ ist darauf zurückzuführen, dass sich die Ersetzungsregel im monografischen Fall für jeden Schlüssel durch ein einzelnes Alphabet charakterisieren lässt
- So wird etwa die Caesarchiffre mit dem Schlüssel  $k = 3$  durch das Alphabet  $\{D, E, F, G, W, \dots, Y, Z, A, B, C\}$  beschrieben
- Dagegen kommen bei der Vigenère-Chiffre periodisch mehrere verschiedene Ersetzungsalphabete zur Anwendung

# Blockchiffren und einfache Substitutionschiffren

Monoalphabetische Chiffrierverfahren ersetzen meist Texteinheiten einer festen Länge  $\ell \geq 1$  durch Kryptotextsegmente der Länge  $\ell$

## Definition

- Sei  $A$  ein beliebiges Alphabet und es gelte  $M = C = A^\ell$ ,  $\ell \geq 1$
- Eine **Blockchiffre** realisiert für jeden Schlüssel  $k \in K$  eine bijektive Abbildung  $g$  auf  $A^\ell$  und es gilt für alle  $x \in M$  und  $y \in C$ ,

$$E(k, x) = g(x) \text{ und } D(k, y) = g^{-1}(y)$$

- Im Fall  $\ell = 1$  spricht man auch von einer **einfachen Substitution**

# Stromchiffren

- Auch polyalphabetische Chiffren verwenden oft eine feste Blocklänge  $\ell$
- Da diese jedoch meist relativ klein ist (meist  $\ell = 1$ ), nennt man die einzelnen Segmente nicht ‚Blöcke‘ sondern ‚Zeichen‘ und spricht von **sequentiellen Chiffren** oder **Stromchiffren**

## Definition

- Sei  $A$  ein beliebiges Alphabet und sei  $M = C = A^\ell$  für eine Zahl  $\ell \geq 1$
- Weiterhin seien  $K$  und  $\hat{K}$  Schlüsselräume
- Eine **Stromchiffre** besteht aus einer Verschlüsselungsfunktion  $E : \hat{K} \times M \rightarrow C$  und einem **Schlüsselstromgenerator**  $g : K \times A^* \rightarrow \hat{K}$
- Der Generator  $g$  erzeugt aus einem **externen** Schlüssel  $k \in K$  für einen Klartext  $x = x_0 \dots x_{n-1}$ ,  $x_i \in M$ , eine Folge  $\hat{k}_0, \dots, \hat{k}_{n-1}$  von **internen** Schlüsseln  $\hat{k}_i = g(k, x_0 \dots x_{i-1}) \in \hat{K}$ , unter denen  $x$  in den Kryptotext

$$E_g(k, x) = E(\hat{k}_0, x_0) \dots E(\hat{k}_{n-1}, x_{n-1})$$

überführt wird

## Stromchiffren

- Der interne Schlüsselraum  $\hat{K}$  einer Stromchiffre kann also wie der Schlüsselraum einer Blockchiffre eine maximale Größe von  $(m^\ell)!$  annehmen
- Im häufigen Spezialfall  $\ell = 1$  hat  $\hat{K}$  also die maximale Größe  $m$ !
- Die Aufgabe des Schlüsselstromgenerators  $g$  ist es, aus dem externen Schlüssel  $k \in K$  und dem bereits verarbeiteten Klartext  $x_0 \dots x_{i-1}$  den aktuellen internen Schlüssel  $\hat{k}_i$  zu berechnen

## Bisher betrachtete Stromchiffren

Chiffre	Chiffrierfunktion	Schlüsselstromgenerator
Vigenère	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \dots k_{d-1}, x_0 \dots x_{i-1}) = k_{(i \bmod m)}$
Beaufort	$E(\hat{k}, x) = \hat{k} - x$	$g(k_0 \dots k_{d-1}, x_0 \dots x_{i-1}) = k_{(i \bmod m)}$
Autokey mit Klartext- Schlüsselstrom	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \dots k_{d-1}, x_0 \dots x_{i-1}) = \begin{cases} k_i, & i < d \\ x_{i-d}, & i \geq d \end{cases}$
Autokey mit Kryptotext- Schlüsselstrom	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \dots k_{d-1}, x_0 \dots x_{i-1}) = \begin{cases} k_i, & i < d \\ y_{i-d}, & i \geq d \end{cases}$ $= k_{(i \bmod d)} + \sum_{j=1}^{\lfloor i/d \rfloor} x_{i-jd}$

- Bei der Vigenère- und Beaufortchiffre hängt der Schlüsselstrom nicht vom Klartext, sondern nur vom externen Schlüssel  $k$  ab, d.h. sie sind **synchrone**
- Die Autokey-Chiffren sind dagegen **asynchron** (und aperiodisch)