

Kryptographie

Fabio Oesch, Michael Künzli & Jan Fässler

4. Semester (FS 2013)

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Mathematische Grundlagen | 1 |
| 1.1 | Modulare Division | 1 |
| 1.2 | Modulares Potenzieren | 1 |
| 1.2.1 | Theorie | 1 |
| 1.2.2 | Beispiel | 1 |
| 2 | Klassische Kryptographie | 3 |
| 2.0 | Repetition | 3 |
| 2.1 | Klassische Verschlüsselungsverfahren | 3 |
| 2.2 | Spezielles Bsp für Substitution Homophone Verschlüsselung | 3 |
| 2.3 | Kasiski-Text (monographisch & polyalphabetisch) | 3 |
| 2.4 | Playfair-Cipher | 4 |
| 2.5 | Koinzidenzindex (index of coincidence) | 4 |
| 2.6 | Vigenères Chipres | 5 |
| 2.6.1 | Berechnung der Schlüssellänge eines Vigenère-Cipher | 5 |
| 2.6.2 | Kryptoanalyse des Vigenère-Cipher | 5 |
| 2.7 | One-Time-Pad | 7 |
| 2.8 | Kryptosysteme | 7 |
| 2.9 | Kryptoanalyse | 7 |
| 2.9.1 | Ciphertext-only attack | 7 |
| 2.9.2 | known-plaintext attack | 7 |
| 2.9.3 | chosen-plaintext attack | 7 |
| 2.9.4 | chosen-ciphertext attack | 8 |
| 3 | Block-Cipher | 9 |
| 3.1 | Data Encryption Standard (DES) | 9 |
| 3.2 | Modi von Block-Cipher | 10 |
| 3.2.1 | ECB-Modus (electronic code block) | 10 |
| 3.2.2 | CBC-Modus (cipher block chaining) | 10 |
| 3.2.3 | CFB-Modus (cipher feedback) | 11 |
| 4 | RSA | 12 |
| 4.1 | Schlüsselerzeugung | 12 |
| 4.2 | Verschlüsselung und Entschlüsselung | 12 |
| 4.2.1 | RSA ist ein Blockcipher | 12 |
| 4.2.2 | Beweis | 12 |

1 Mathematische Grundlagen

1.1 Modulare Division

Eine modulare Division hat die Form $\boxed{a/b \bmod n}$, gesucht wird die ganze Zahl c im Intervall $[0, n-1]$, welche die Gleichung $\boxed{bc \equiv a \bmod n}$.

Die modulare Division ist nur möglich, wenn $\text{ggT}(b, n) = 1$.

Beispiel: $23/27 \bmod 31$

Zuerst $\text{ggT}(27, 31)$ mittels euklidischem Algorithmus ermitteln:

$$31 = 1 * 27 + 4$$

$$27 = 6 * 4 + 3$$

$$4 = 1 * 3 + 1$$

$$3 = 3 * 1 + 0 \implies \text{ggT}(27, 31) = 1 \rightarrow \text{modulare Division möglich}$$

Jetzt fahren wir mit dem erweiterten euklidischen Algorithmus fort, um c zu ermitteln. Dafür müssen wir zuerst die lineare diophantische Gleichung $23 = 27c + 31x$ lösen:

$$1 = 4 - 1 * 3$$

$$1 = 4 - 1 * (27 - 6 * 4) // \text{ ersetze 3 durch diese Klammer, indem man obigen Algorithmus rückwärts durchläuft}$$

$$1 = 4 - 1 * 27 + 6 * 4 = 7 * 4 - 1 * 27 // \text{ ausmultiplizieren}$$

$$1 = 7 * (31 - 1 * 27) - 1 * 27 // \text{ ersetze 4 durch Klammer}$$

$$1 = 7 * 31 - 7 * 27 - 1 * 27 = 7 * 31 - 8 * 27 // \text{ ausmultiplizieren}$$

$$23 * 1 = 23 * 7 * 31 + 23 * (-8) * 27 // \text{ erweitern mit 23}$$

\implies uns interessiert nur $c = 23 * (-8) = -184$ was der **Restklasse 2** (von Modulo 31) entspricht. Dies ermittelt man, indem man zu -184 so oft 31 addiert, bis man eine positive Zahl erhält.

Die gesuchte Gleichung lautet also: $27 * 2 \equiv 23 \bmod 31$.

1.2 Modulares Potenzieren

1.2.1 Theorie

Seien $a, b, n \in \mathbb{Z}$ und $b, n > 1$. Berechnen Sie $a^b \bmod n$.

Da es für grosse b für den Taschenrechner nicht möglich ist dies zu berechnen verwenden wir ein spezielles Verfahren:

- 1.) binäre Darstellung von b :

$$b = \sum_{i=0}^k \alpha_i 2^i \text{ mit } \alpha \in \{0, 1\}.$$

- 2.) Anwendung auf a :

$$a^b = a^{\sum_{i=0}^k \alpha_i 2^i}$$

$$a^b = \prod_{i=0}^k a^{\alpha_i 2^i}$$

$$a^b = a^{\alpha_k 2^k} * a^{\alpha_{k-1} 2^{k-1}} * a^{\alpha_{k-2} 2^{k-2}} \dots a^{\alpha_1 2} * a^{\alpha_0}$$

$$a^b = (\dots ((a^{\alpha_k})^2 * a^{\alpha_{k-1}})^2 \dots * a^{\alpha_1})^2 * a^{\alpha_0}$$

- 3.) Das Verfahren besteht nun darin, den letzten Ausdruck von innen nach aussen auszuwerten und nach jeder Multiplikation das Resultat modulo n zu rechnen.

1.2.2 Beispiel

$$977^{2222} \bmod 11$$

- 1.) $2222_{10} \blacktriangleright \text{bin} = 100010101110_2$

- 2.) $(\dots ((977^2)^2)^2 * 977)^2 * 977)^2 * 977)^2 * 977)^2 * (0 * 977)$

3.) Anwendung des Verfahren:

| | | |
|-----------|--------|------------|
| 977 | mod 11 | = 9 |
| 9^2 | mod 11 | = 4 |
| 4^2 | mod 11 | = 5 |
| 5^2 | mod 11 | = 3 |
| 3^2 | mod 11 | = 9 |
| $9 * 977$ | mod 11 | = 4 |
| 4^2 | mod 11 | = 5 |
| 5^2 | mod 11 | = 3 |
| $3 * 977$ | mod 11 | = 5 |
| 5^2 | mod 11 | = 3 |
| 3^2 | mod 11 | = 9 |
| $9 * 977$ | mod 11 | = 4 |
| 4^2 | mod 11 | = 5 |
| $5 * 977$ | mod 11 | = 1 |
| 1^2 | mod 11 | = 1 |
| $1 * 977$ | mod 11 | = 9 |
| 9^2 | mod 11 | = 4 |

2 Klassische Kryptographie

2.0 Repetition

Alphabet endliche Mengen von Zeichen

Beispiel

$$\mathcal{A} := \{A, B, C, \dots, Z\}, |\mathcal{A}| = 26$$

$$\Sigma := \{0, 1\}, |\Sigma| = 2$$

$$\mathcal{A}^* := \{\text{endliche Wörter über } \mathcal{A}\}$$

Sprachen über \mathcal{A} : $L \subset \mathcal{A}^*$

2.1 Klassische Verschlüsselungsverfahren

| Substitution Cipher | Transposition Cipher | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Einheiten werden ersetzt . | Einheiten werden vertauscht . | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><td>3</td><td>1</td><td>5</td><td>6</td><td>2</td><td>4</td></tr><tr><td>K</td><td>O</td><td>M</td><td>M</td><td>E</td><td>H</td></tr><tr><td>E</td><td>U</td><td>T</td><td>E</td><td>A</td><td>B</td></tr><tr><td>E</td><td>N</td><td>D</td><td>Z</td><td>U</td><td>M</td></tr><tr><td>Z</td><td>O</td><td>O</td><td>A</td><td>B</td><td>C</td></tr></table> | 3 | 1 | 5 | 6 | 2 | 4 | K | O | M | M | E | H | E | U | T | E | A | B | E | N | D | Z | U | M | Z | O | O | A | B | C |
| 3 | 1 | 5 | 6 | 2 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| K | O | M | M | E | H | | | | | | | | | | | | | | | | | | | | | | | | | | |
| E | U | T | E | A | B | | | | | | | | | | | | | | | | | | | | | | | | | | |
| E | N | D | Z | U | M | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Z | O | O | A | B | C | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | $\Rightarrow \underbrace{\text{OUNO}}_1 \underbrace{\text{EAUB}}_2 \dots$ Bem. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Einheiten werden vertauscht (ABC ist Padding) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| monoalphabetisch $E : \mathcal{A} \rightarrow B, x \mapsto E(x)$ | polyalphabetisch $E : \mathcal{A} \rightarrow P(B), x \mapsto E(x)$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| monographisch Buchstaben | polygraphisch Gruppen von Buchstaben | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

2.2 Spezielles Bsp für Substitution Homophone Verschlüsselung

Gegeben: $\Sigma := \{0, 1\}, B := \{a, b, c\}$

Information über die Sprache des Klartextes: Häufigkeit von 0 : $\frac{1}{3}$
Häufigkeit von 1 : $\frac{2}{3}$

$$E : \Sigma \rightarrow P(B)$$

$$0 \mapsto \{b\}$$

$$1 \mapsto \{a, c\}$$

Bsp: 10110110011
abccbacbbaa

2.3 Kasiski-Text (monographisch & polyalphabetisch)

Klartext TO BE OR NOT TO BE

Schlüssel NOW

$$\mathbf{p} = |\text{NOW}|$$

| | | | | |
|-----|-----|-----|-----|---|
| TOB | EOR | NOT | TOB | E |
| NOW | NOW | NOW | NOW | N |
| GCX | RCN | ACP | GCX | R |

GCX kommt 2x for so können wir eine Annahme zur Periode p machen. Die Periode ist dann $c \cdot p$. Dies kann aber auch zufällig passieren.

2.4 Playfair-Cipher

| | | | | | | | | |
|---|---------------------------------------|----|----|----|----|-----|----|----|
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> HARYP OTEBC DFG^I_JK LMNQS UVWXZ </div> | Schlüssel: Harry Potter, HARRY POTTER | | | | | | | |
| | Klartext | HA | LL | O | ZU | SA | MM | EN |
| | Bsp: Preprocessed | HA | LX | LO | ZU | SA | MX | NX |
| | Secret | AR | QU | UD | UV | ... | | |
| | | | | | | | | |

- Falls 2 auf gleicher Zeile: Beide Buchstaben um eins nach rechts
- Falls 2 auf gleicher Spalte: Beide Buchstaben um eins nach unten
- Falls 2 nicht auf gleicher Zeile/Spalte: Man nimmt die Buchstaben die auf seiner Spalte und auf der anderen Zeile liegen.

| | | | |
|---|---|---|---|
| L | M | N | Q |
| ↓ | | | ↑ |
| U | V | W | X |

2.5 Koinzidenzindex (index of coincidence)

1. Gegeben

Alphabet Alphabet $\mathcal{A} := \{A, B, C, \dots, Z\}$
Sprache: Englisch

\Rightarrow Buchstabenhäufigkeit:

| | | | |
|-------|-------|---------|-------|
| p_A | p_B | \dots | p_Z |
| p_1 | p_2 | \dots | p_3 |

mit $0 \leq p_i \leq 1$ und $\sum_{i=1}^{26} p_i = 1$

IC: Grösse, die von der Sprache abhängt, aber invariant ist gegenüber Cäsar-Verschiebungen.

Frage: Was bedeutet: Was bedeutet $IC_L := \sum_{i=1}^{26} P_i^2$ index of coincidence L: Language

Bemerkung:

Jede Sprache hat ihren eigenen Koinzidenzindex

$IC_{German} = 0.0766$

$IC_{Arabic} = 0.0759$

$IC_{flat} = 0.0385$ (Alle Buchstaben haben die gleiche Häufigkeit: $p_1 = p_2 = \dots = p_{26} = \frac{1}{26}$)

Je unregelmässiger die Buchstabenhäufigkeit, umso grösser der Index.

2. Gegeben:

Sei F eine Buchstabenfolge der Länge n

Bsp: $F = "AXCAABCXA"$

$n_1 = \#A's \text{ in } F$

$n_2 = \#B's \text{ in } F$

\vdots

Frage: Wie gross ist die Wahrscheinlichkeit zwei gleiche Buchstaben aus F herauszugreifen?

Definition $IC_F = \frac{\sum_1^{26} \binom{n_i}{2}}{\binom{n}{2}}$

Bsp:

Alphabet $\Sigma := \{0, 1\}$

F = 00110111101

$$\left. \begin{array}{l} n_0 = 4 \\ n_1 = 7 \\ n = 11 \end{array} \right\} IC_F = \frac{4 \cdot 3 + 7 \cdot 6}{11 \cdot 10} = 0.49$$

Annahme $IC_F \xrightarrow{F \rightarrow \infty} IC_L$ (ist im Allgemeinen falsch)

Bemerkung

Permutation der Buchstaben

$F \mapsto \text{Perm}(F)$

$F = \text{"AXCA..."} \mapsto \text{Perm}(F) = \text{"CBYC..."}$

$$IC_F = IC_{\text{Perm}(F)}$$

2.6 Vigenères Chipres

2.6.1 Berechnung der Schlüssellänge eines Vigenère-Cipher

Gegeben

C Vigenère-Chiffre der Länge n

Die Schlüssellänge sei p (unbekannt)

| $\overbrace{\hspace{10em}}^p$ | | | | | |
|-------------------------------|------------|------------|------------|---------|----------|
| C_1 | C_2 | C_3 | C_4 | \dots | C_p |
| C_{p+1} | C_{p+2} | C_{p+3} | C_{p+4} | \dots | C_{2p} |
| C_{2p+1} | C_{2p+2} | C_{2p+3} | C_{2p+4} | \dots | C_{3p} |
| \dots | \dots | \dots | \dots | \dots | \dots |
| C_{n-2} | C_{n-1} | C_n | - | - | - |

↑

monoalphabetisch

alle Spalten = p, alle Zeilen = $\frac{n}{p}$, letzte Zeile = monoalphabetisch!

$\alpha :=$ Anzahl Buchstabenpaare aus gleicher Spalte, $\alpha = \frac{n(\frac{n}{p}-1)}{2} = \frac{n(n-p)}{2p}$

$\beta :=$ Anzahl Buchstabenpaare aus verschiedenen Spalten, $\beta = \frac{n(n-\frac{n}{p})}{2} = \frac{n^2(p-1)}{2p}$

$\gamma :=$ Anzahl gleicher Buchstabenpaare aus C, $IC_L = \frac{\gamma}{\binom{n}{2}}$

$$\gamma = \alpha \cdot IC_L + \beta \cdot IC_{\text{flat}}$$

$$p = \frac{n(IC_L - IC_{\text{flat}})}{IC_C \cdot (n-1) + IC_L - n \cdot IC_{\text{flat}}}$$

2.6.2 Kryptoanalyse des Vigenère-Cipher

1) Schlüssellänge p

p=1,2,3,...

- Einleitung des Cipher-Tests in p Abschnitte
- Berechnung des IC des Abschnitts
- Wähle p mit $IC \sim IC_2$ (oder hoch)

2) Sei s,t zwei Strings über dem Alphabet A.

$s = s_1, s_2, s_3, \dots, s_k$

$$t = t_1, t_2, t_3, \dots, t_l$$

Wieder zählen wir $n_1(s) := A$ in s , $n_3(t) = C$ in t

Def. $MIC(s, t) := \frac{\sum_{i=1}^{26} n_i(s) * n_i(t)}{k * l}$

Bsp.

$s = \text{"AABCCA"}$

$t = \text{"ÄBCABCABC"}$

$$n_1(s) = 3, n_1(t) = 3$$

$$n_2(s) = 1, n_2(t) = 3$$

$$n_3(s) = 2, n_3(t) = 3$$

$$\rightarrow MIC(s, t) = \frac{1}{6 * 9} [3 * 3 + 1 * 3 + 2 * 3]$$

Idee: s, t zwei cipher-Text mit Cäsar Cerschlüsselung

Wenn beide mit dem gleichen Schlüssel verschlüsselt werden

$$\rightarrow MIC(s, t) \rightsquigarrow IC_L$$

$$\text{Sonst: } MIC(s, t) \rightsquigarrow IC_{flat}$$

3.) Anwendung auf Cipher Text

Schlüssellänge p sei 5

c_1, c_2, \dots, c_5 Abschnitte des Cipher Text

$$MIC(c_i, c_j + k)$$

Tabelle:

| $(i, j); k$ | 0 | 1 | 2 | ... |
|-------------|---|---|-----|-----|
| (1, 2) | | | | |
| (1, 3) | | | | |
| (1, 4) | | | | |
| (1, 5) | | | | |
| (2, 3) | | | x | |
| (2, 4) | | | | |
| (2, 5) | | | | |
| (3, 4) | | | | |
| (3, 5) | | | | |
| (4, 5) | | | | |

$$\rightarrow MIC(c_2, c_3 + k)$$

Bsp

c_1 : AXBM...

c_3 : ABXHE...

$c_3 + 2$: CDZJG

4.) Wir suchen Einträge in der Tabelle, die hoch sind (> 0.06)

$$MIC(s, t) = \frac{1}{kl} \sum_{i=1}^{26} n_i(s) n_i(t), |s| = k, |t| = l$$

$$\text{zb: } MIC(c_2, c_3 + 22) > 0.06 \iff c_2 \sim c_3 + 22 \Rightarrow \boxed{\beta_2 - \beta_3 = k}$$

Notation $s \sim t \iff s$ und t sind mit dem gleichen Shift aus zwei Klartexten entstanden.

Bsp. $klar_1 \sim klar_2$

$$klar_1 \xrightarrow{\beta_1} c_1 \quad \left| \quad c_1 = klar_1 + \beta_1 \right.$$

$$klar_2 \xrightarrow{\beta_2} c_2 \quad \left| \quad c_2 = klar_2 + \beta_2 \right.$$

Wir suchen die grossen Werte von $MIC(c_i, c_j + k)$

$$MIC(c_i, c_j + k) \text{ gross} \iff c_i \sim c_j + k$$

$$c_i = klar_i + \beta_i \sim klar_i + \beta_j + k = \textcolor{red}{k} = \textcolor{red}{\beta_i + \beta_j}$$

\downarrow sind bekannt
 $\left. \begin{array}{l} k_{12} = \beta_2 - \beta_1 \\ k_{13} = \beta_3 - \beta_1 \\ k_{52} = \beta_2 - \beta_5 \end{array} \right\} \text{Auflösen nach } \beta_1$
Schlüsselwort: $\beta_1, \beta_2, \dots, \beta_p = \beta_1, \beta_1 + k_{12}, \dots,$
Ausprobieren: $\beta_1 = 0, 1, \dots, 25$

2.7 One-Time-Pad

$\Sigma = \{0, 1\}$

| | | | |
|-------------|--|---|---------|
| Klartext: | $p_1 p_2 p_3 p_4 p_5 \dots =$ | 0 | 0101... |
| Schlüssel: | $k_1 k_2 k_3 k_4 k_5 \dots =$ | 1 | 0110... |
| ciphertext: | $\begin{array}{l} c_1 \quad c_2 c_3 c_4 c_5 \dots = \\ p_1 \oplus k_1 \end{array}$ | 1 | 0011... |

2.8 Kryptosysteme

Kryptosystem: (P, C, K, e, d)

P Menge der **Klartexte**

C Menge der **Geheimtexte**

K Menge der Schlüssel

$$e : K \times P \rightarrow C$$

$$d : K \times C \rightarrow P$$

$\forall k \in K \quad \forall p \in P : d(k, e(k, p)) = p$
 $\rightarrow \forall k \in K : e(k, -)$ ist **injektiv**
 $\rightarrow \forall k \in K : d(k, -)$ ist **surjektiv**

2.9 Kryptoanalysis

2.9.1 Ciphertext-only attack

Gegeben $c_i = e_k(p_i), i=1, \dots, n$

Gesucht $p_i, i=1, \dots, n$ oder k

2.9.2 known-plaintext attack

Gegeben $(p_i, c_i = e_k(p_i)), i=1, \dots, n$

Gesucht k

2.9.3 chosen-plaintext attack

Gegeben $(p_i, c_i = e_k(p_i)), i=1, \dots, n$
 p_i nach Wahl des Kryptoanalytikers

Gesucht k

Verwendung DIE Attacke gegen jedes Public-Key System

2.9.4 chosen-ciphertext attack

Gegeben $(p_i, p_i = d_k(c_i)), i=1, \dots, n$
 c_i nach Wahl des Kryptoanalytikers

Gesucht k

3 Block-Cipher

Alphabet

$$\Sigma = \{0, 1\}$$

$$\Sigma^n := \Sigma \times \Sigma \times \dots \times \Sigma$$

Definition

Ein Block - Cipher ist eine **injektive** Abbildung

$$C : K \rightarrow \text{Perm}(\Sigma^n)$$

wobei K der Schlüsselraum ist.

Bsp.

$$n = 3$$

$$\Sigma^3 = \Sigma \times \Sigma \times \Sigma$$

$$p \left\{ \begin{array}{ccc} 000 & \nearrow & 000 \\ 001 & \rightarrow & 001 \\ \vdots & & \vdots \\ 111 & \searrow & 111 \end{array} \right\} l$$

↑ Schlüssel

Frage:

Wie gross ist der Schlüsselraum K maximal?

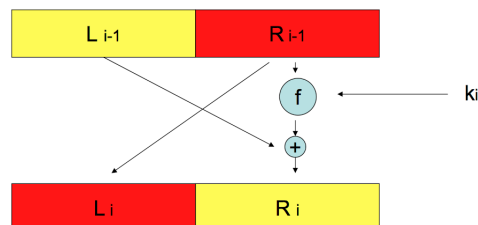
$$|K| \leq (2^n)!$$

3.1 Data Encryption Standard (DES)

Lucifer Schlüssellänge 128

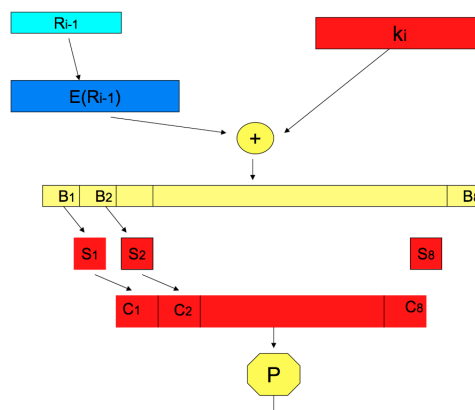
↓

DES Schlüssellänge 56
 Blocklänge 64



$$\begin{aligned} L_1 &:= R_0 \\ R_1 &:= f(R_0, k_1) \oplus L_0 \\ L_0 &:= f(L_1, k_1) \oplus R_1 \\ R_0 &:= L_1 \end{aligned}$$

Die f-Funktion:



3.2 Modi von Block-Cipher

Sei $\Sigma := \{0, 1\}$

$p = c = \Sigma^4 = \{\square\square\square\square\}$

$k = \text{Permutation von } \Sigma^4$

$$k = \pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

Vor- und Entschlüsselung

Sei $m = 0101 \in p$ (Klartext)

$$e_k(m) = e_k(0101) = 1010 = c$$

3.2.1 ECB-Modus (electronic code block)

$$m = \underbrace{1100}_{m_1} | \underbrace{0110}_{m_2} | \underbrace{1100}_{m_3} | 101^*$$

$$\xrightarrow[m_1]{\boxed{e_k}} c_1$$

Bem:

1. $m_1 = m_3 \Rightarrow c_1 = c_3$
2. Vertauschen der Ciphertext-Blöcke wird nicht notwendigerweise erkannt

3.2.2 CBC-Modus (cipher block chaining)

$$m = \underbrace{m_1}_{\text{Länge } n} | m_2 | \dots, n : \text{Blocklänge}$$

$$\text{Bsp: } m = \underbrace{1100}_{m_1} | \underbrace{0110}_{m_2} | \underbrace{1100}_{m_3} | 101$$

$$IV = C_0 = 1110$$

IV = Initialvektor (i.a. bekannt)

$$C_0 := IV$$

$$C_1 := e_k(C_0 \oplus m_1)$$

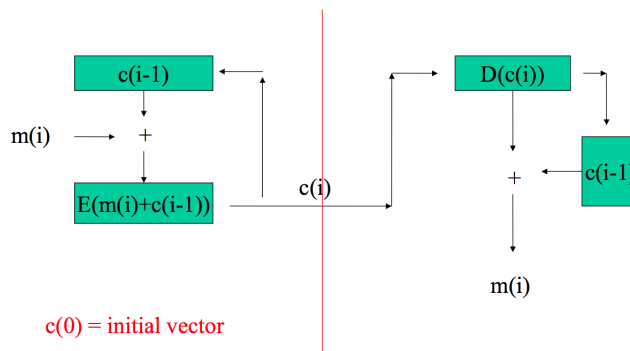
$$C_2 := e_k(C_1 \oplus m_2)$$

$$c_1 = e_k(c_0 \oplus m_1) = e_k(0010) = 0001$$

$$c_2 = e_k(c_1 \oplus m_2) = e_k(0111) = 1011$$

$$c_3 = e_k(c_2 \oplus m_3) = e_k(0111) = 1011$$

Entschlüsselung: $c_1 \oplus d_k(c_2) = c_1 \oplus d_k(e_k(c_1 \oplus m_2)) = c_1 \oplus m_2 \oplus c_1 = m_2$



$$m = \underbrace{m_1}_{\text{Länge } n} | m_2, n : \text{Blocklänge}$$

IV = Initialvektor (i.a. bekannt)

$$c_0 := IV, c_1 := e_k(c_0 \oplus m_1), c_2 := e_k(c_1 \oplus m_2)$$

$$c_1 \oplus d_k(c_2) = d_k(e_k(c_1 \oplus m_2)) = c_1 \oplus m_2 \oplus c_1 = m_2$$

$$\text{Bsp: } m = \underbrace{1100}_{m_1} | \underbrace{0110}_{m_2} | \underbrace{1100}_{m_3} | 101, IV = c_0 = 1110$$

$$c_1 = e_k(c_0 \oplus m_1) = e_k(0010) = 0001$$

$$c_2 = e_k(c_1 \oplus m_2) = e_k(0111) = 1011$$

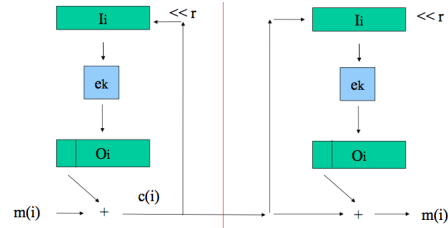
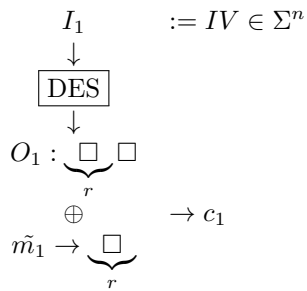
$$c_3 = e_k(c_2 \oplus m_3) = e_k(0111) = 1011$$

Bem:

1. $m_1 = m_3 \nRightarrow c_1 = c_3$
2. Vertauschen kann bemerkt werden
3. Übertragungsfaktor machen sich bemerkbar

3.2.3 CFB-Modus (cipher feedback)

$$m = \underbrace{\tilde{m}_1}_{\text{Länge}=r} | \tilde{m}_2 | \tilde{m}_3 | \dots, n: \text{ Cipher Block-Länge (DES: 64) und } \boxed{0 < r \leq n}$$



Bsp: $m = 110|001|101|100|101$, $IV = 1110$, $\boxed{r = 3, n = 4}$

| | |
|---|---|
| $I_1 =$ \downarrow $\boxed{e_k}$ \downarrow $O_1 = 1101$ \oplus $\tilde{m}_1 = 110 \rightarrow c_1 = 000$ | $I_2 =$ \downarrow $\boxed{e_k}$ \downarrow $O_2 = 0000$ \oplus $\tilde{m}_2 = 001 \rightarrow c_2 = 001$ |
|---|---|

4 RSA

4.1 Schlüsselerzeugung

PK = (n,e)

SK = (n,d)

Wir wählen zwei (grosse) Primzahlen $p, q \in \mathbb{R}^*$. $\varphi \neq q$

$n = p * q$

$\varphi(n) = (p-1)(q-1) // \varphi(n) = |\mathbb{Z}_n^*|$

Wir wählen $e \in \mathbb{Z}_{\varphi(n)}^*$ // $\text{ggT}(e, \varphi(n)) = 1$

$d := e^{-1} \text{ in } \mathbb{Z}_{\varphi(n)}^*$ // $ed=1 \text{ in } \mathbb{Z}_{\varphi(n)}^* \Leftrightarrow ed \equiv 1 \pmod{\varphi(n)}$

$\Rightarrow \varphi(n) | (ed - 1)$

$\Rightarrow \boxed{\exists k \in \mathbb{Z} : e * d + k * \varphi(n) = 1}$

$d := e^{-1} \in \mathbb{Z}_{120}^* : \boxed{ed + k\varphi(n) = 1}$

Beispiel:

$p = 11, q = 13$

$n = p * q = 143$

$\varphi(n) = 120 = 2^3 * 3 * 5$

$e := 7 \Rightarrow \text{PK} = (143, 7)$

$\mathbb{Z}_n = \{0, 1, 2, 3, \dots, n-1\}$

| i | q_i | r_i | s_i | t_i |
|---|-------|-------|-------|-------|
| 0 | - | 120 | 1 | 0 |
| 1 | 17 | 7 | 0 | 1 |
| | | 1 | 1 | -17 |

$120 = q * 7 + r$

$\Rightarrow (*) \underbrace{e}_7 * (-17) + 1 * \underbrace{\varphi(n)}_{120} = 1 // \pmod{\varphi(n)} \Rightarrow \boxed{d \equiv (-17) \pmod{\varphi(n)}}$

4.2 Verschlüsselung und Entschlüsselung

4.2.1 RSA ist ein Blockcipher

encryption : enc

$\mathbb{Z}_n \longrightarrow \mathbb{Z}_n$

$m \longrightarrow c^e \pmod{n}$

decryption : dec

$\mathbb{Z}_n \longrightarrow \mathbb{Z}_n$

$m \longrightarrow c^d \pmod{n}$

$\left. \begin{array}{l} PK = (u, e) \\ SK = (u, d) \end{array} \right\} \boxed{\forall m \in \mathbb{Z}_n : \text{dec}_{SK}(\text{enc}_{PK}(m)) = m}$

4.2.2 Beweis

Fall 1:

$\text{ggT}(m, n) = 1$

$(m^e)^d = m \text{ in } \mathbb{Z}_n$

Weil $\text{ggT}(m, n) = 1$ existiert das Inverse von m: $\underbrace{m^{ed-1}} = 1 \text{ in } \mathbb{Z}_n$

Das ist zu Zeigen!

$$e * d + k * \varphi(n) = 1 \text{ // Konstruktion des Schlüssels}$$

$$\Rightarrow e * d - 1 = -k * \varphi(n) : m^{ed-1} = m^{-k * \varphi(n)} = (m^{-k}) = 1 \text{ // Satz von Euler-Fermat}$$

Fall 2:

$$\text{ggT}(m, n) \neq 1 \Rightarrow m = l * p \text{ oder } m = k * q$$