Kryptographie

Fabio Oesch, Michael Künzli & Jan Fässler

4. Semester (FS 2013)

Inhaltsverzeichnis

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 $ggT(b, n) = 1$.

Beispiel: $23/27 \bmod 31$

Zuerst $ggT(27, 31)$ mittels euklidischem Algorithmus ermitteln:
$31 = 1 * 27 + 4$
$27 = 6 * 4 + 3$
$4 = 1 * 3 + 1$
$3 = 3 * 1 + 0 \Longrightarrow ggT(27, 31) = 1 \rightarrow$ 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:

$1 = 4 - 1 * 3$
$1 = 4 - 1 * (27 - 6 * 4)$ // ersetze 3 durch diese Klammer, indem man obigen Algorithmus rückwärts durchläuft
$1 = 4 - 1 * 27 + 6 * 4 = 7 * 4 - 1 * 27$ // ausmultiplizieren
$1 = 7 * (31 - 1 * 27) - 1 * 27$ // ersetze 4 durch Klammer
$1 = 7 * 31 - 7 * 27) - 1 * 27 = \mathbf{7} * 31$ **-8** $* 27$ // ausmultiplizieren
$23 * 1 = 23 * \mathbf{7} * 31 + 23 *$ **(-8)** $* 27$ // erweitern mit 23
$\Longrightarrow$ 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:
$b = \sum_{i=0}^{k} \alpha_i 2^i$ mit $\alpha \in \{0, 1\}$.

2.) Anwendung auf a:
$a^b = a^{\sum_{i=0}^{k} \alpha_i 2^i}$
$a^b = \prod_{i=0}^{k} a^{\alpha_i 2^i}$
$a^b = a^{\alpha_k 2^k} * a^{\alpha_{k-1} 2^{k-1}} * a^{\alpha_{k-2} 2^{k-2}} \ldots a^{\alpha_1 2} * a^{\alpha_0}$
$a^b = (\ldots((a^{a_k})^2 * a^{a_{k-1}})^2 \ldots * a^{\alpha_1})^2 * a^{\alpha_0}$

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} ► bin $= 100010101110_2$

2.) $(\ldots(977)^2)^2)^2)^2 * 977)^2)^2 * 977)^2)^2 * 977)^2 * 977)^2 * 977)^2 * (0 * 977)$

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	$= \mathbf{4}$

2 Klassische Kryptographie

2.0 Repetition

Alphabet endliche Mengen von Zeichen

Beispiel

$\mathcal{A} := \{A, B, C, ..., Z\}$, $|\mathcal{A}| = 26$
$\Sigma := \{0, 1\}$, $|\Sigma| = 2$
$\mathcal{A}^* := \{\text{endliche Wörter über } \mathcal{A}\}$

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

2.1 Klassische Verschlüsselungsverfahren

Substitution Cipher	**Transposition Cipher**
Einheiten werden **ersetzt**.	Einheiten werden **vertauscht**. $\begin{array}{cccccc} 3 & 1 & 5 & 6 & 2 & 4 \\ \hline K & O & M & M & E & H \\ E & U & T & E & A & B \\ E & N & D & Z & U & M \\ Z & O & O & A & B & C \end{array}$ $\Rightarrow \underbrace{\text{OUNO}}_{1}\underbrace{\text{EAUB}}_{2}\ldots$ **Bem.** Einheiten werden vertauscht (ABC ist Padding)

monoalphabetisch	**polyalphabetisch**
$E : \mathcal{A} \to B$, $x \mapsto E(x)$	$E : \mathcal{A} \to P(B)$, $x \mapsto E(x)$
monographisch	**polygraphisch**
Buchstaben	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 : $\frac{2}{3}$

$$\begin{aligned} E :& \Sigma \to P(B) \\ 0 &\mapsto \{b\} \\ 1 &\mapsto \{a, c\} \end{aligned}$$

Bsp: 10110110011
abccbacbbaa

2.3 Kasiski-Text (monographisch & polyalphabetisch)

Klartext TO BE OR NOT TO BE

Schlüssel NOW

p = |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

$\boxed{\begin{array}{c}\text{HARYP}\\ \text{OTEBC}\\ \text{DFG}\tfrac{\text{I}}{\text{J}}\text{K}\\ \text{LMNQS}\\ \text{UVWXZ}\end{array}}$ Schlüssel: Harry Potter, HAR~~R~~Y POT~~T~~E~~R~~

	Klartext	HA	LL	O	ZU	SA	MM	EN
Bsp:	Preprocessed	HA	LX	LO	ZU	SA	MX	ME	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 des anderen Zeile liegen.

  $\begin{array}{cccc} \text{L} & \text{M} & \text{N} & \text{Q} \\ \downarrow & & & \uparrow \\ \text{U} & \text{V} & \text{W} & \text{X} \end{array}$

2.5 Koinzidenzindex (index of coincidence)

1. **Gegeben**

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

   $\Rightarrow$ Buchstabenhäufigkeit: $\begin{array}{cccc} p_A & p_B & \cdots & p_Z \\ \shortparallel & \shortparallel & & \shortparallel \\ p_1 & p_2 & \cdots & p_3 \end{array}$

   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 $\boxed{IC_L := \sum_{1=1}^{26} P_i^2}$ **i**ndex of **c**oincidence L: Language

   Bemerkung:

   Jede Sprache hat ihren eigenen Konzidenzindex
   $IC_{German} = 0.0766$
   $IC_{Arabic} = 0.0759$
   $IC_{flat} = 0.0385$ (Alle Buchstaben haben die gleiche häufigkeit: $p_1 = p_2 = ... = p_{26} = \frac{1}{26}$)
   Je unregelmässiger die buchstabenhäufigkeit, umso grösser der Index.

2. **Gegegen:**

   Sei F eine Buchstabenfolge der Länge n
   Bsp: $F =$"$AXCAABCXA$"
   $n_1 = \#A's$ in F
   $n_1 = \#B's$ in F
   $\vdots$

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

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

Bsp:

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

F = 00110111101

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

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

Bemerkung

Permutation der Buchstaben

F $\mapsto$ Perm(F)

F = "AXCA…"$\mapsto$ Perm(F) = "CBYC…"

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

2.6 Vigenères Chipres

2.6.1 Beschreibung

Die im 16. Jahrhundert entstandene Vigenère-Verschlüsselung galt lange als sicherer Chiffrieralgorithmus. Ein Schlüsselwort bestimmt, wie viele und welche Alphabete genutzt werden. Die Alphabete leiten sich aus der Caesar-Substitution ab.

2.6.2 Beispiel

Das Schlüsselwort sei „AKEY“, der Text „geheimnis“. Vier Caesar-Substitutionen verschlüsseln den Text. Die erste Substitution ist eine Caesar-Verschlüsselung mit dem Schlüssel „A“. „A“ ist der erste Buchstabe im Alphabet. Er verschiebt den ersten Buchstaben des zu verschlüsselnden Textes, das „g“, um 0 Stellen, es bleibt „G“. Der zweite Buchstabe des Schlüssels, das „K“, ist der elfte Buchstabe im Alphabet, er verschiebt das zweite Zeichen des Textes, das „e“, um zehn Zeichen. Aus „e“ wird ein „O“ (siehe Tabelle). Das dritte Zeichen des Schlüssels („E“) verschiebt um 4, „Y“ um 24 Stellen. Die Verschiebung des nächsten Buchstabens des Textes beginnt wieder bei „A“, dem ersten Buchstaben des Schlüssels:

Klartext:	g	e	h	e	i	m	n	i	s
Schlüssel:	A	K	E	Y	A	K	E	Y	A
Geheimtext:	G	O	L	C	I	W	R	G	S

2.6.3 Tabelle

2.6.4 Berechnung der Schlüssellänge eines Vigenère-Cipher

Gegeben

C Vigenère-Chiffrat der Länge n

Die Schlüssellänge sei p (unbekannt)

$$\overbrace{\begin{array}{|c|c|c|c|c|c|} \hline C_1 & C_2 & C_3 & C_4 & \cdots & C_p \\ \hline C_{p+1} & C_{p+2} & C_{p+3} & C_{p+4} & \cdots & C_{2p} \\ \hline C_{2p+1} & C_{2p+2} & C_{2p+3} & C_{2p+4} & \cdots & C_{3p} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline C_{n-2} & C_{n-1} & C_n & - & - & - \\ \hline \end{array}}^{p} \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\end{array}}\right\} \frac{n}{p}$$

$\uparrow \qquad \nearrow \qquad \nearrow$

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, $IC_L = \frac{\gamma}{\binom{n}{2}}$

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

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

2.6.5 Kryptoanalysis 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.

$s = s_1, s_2, s_3, ...s_k$

$t = t_1, t_2, t_3, ..., t_l$

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=”AABCCA”

t=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

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

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

3.) Anwendung auf Cipher Text

Schlüssellänge p sei 5

$c_1, c_2, ..., c_5$ Abschnitte des Cipher Text

$MIC(c_i, c_j + k)$

Tabelle:

$(i,j);k$	0	1	2	$\cdots$
$(1,2)$				
$(1,3)$				
$(1,4)$				
$(1,5)$				
$(2,3)$			x	
$(2,4)$				
$(2,5)$				
$(3,4)$				
$(3,5)$				
$(4,5)$				

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

Bsp

c_1:	AXBM...
c_3:	ABXHE...
$c_3 + 2$:	CDZJG

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

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

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

Bsp. $klar_1 \sim klar_2$

$$\begin{array}{l|l} klar_1 \xrightarrow{\beta_1} c_1 & c_1 = klar_1 + \beta_1 \\ klar_2 \xrightarrow{\beta_2} c_2 & c_2 = klar_2 + \beta_2 \end{array}$$

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

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

$c_i = klar_i + \beta_i \sim klar_i + \beta_j + k = k = \beta_i + \beta_j$

$\downarrow$ sind <u>bekannt</u>

$$\left.\begin{array}{l} k_{12} = \beta_2 - \beta_1 \\ k_{13} = \beta_3 - \beta_1 \\ k_{52} = \beta_2 - \beta_5 \end{array}\right\} \text{Auflösen nach } \beta_1$$

Schlüsselwort: $\beta_1, \overset{\text{abhängig von } \beta_1}{\beta_2, \ldots, \beta_p} = \beta_1, \beta_1 + k_{12}, \ldots,$

Ausprobieren: $\beta_1 = 0, 1, \ldots, 25$

2.7 One-Time-Pad

$$\Sigma = \{0,1\} \quad \begin{array}{ll} \text{Klartext:} & p_1 p_2 p_3 p_4 p_5 \cdots = \\ \text{Schlüssel:} & k_1 k_2 k_3 k_4 k_5 \cdots = \\ \text{ciphertext:} & \underset{p_1 \oplus k_1}{c_1}\ c_2 c_3 c_4 c_5 \cdots = \end{array} \boxed{\begin{array}{c} 0 \\ 1 \\ 1 \end{array}} \begin{array}{l} 0101\ldots \\ 0110\ldots \\ 0011\ldots \end{array}$$

2.8 Kryptosysteme

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

P Menge der Klartexte

C Menge der Geheimtexte

K Menge der Schlüssel

$e : K \times P \to C$

$d : K \times C \rightarrow P$

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

2.9 Kryptoanalysis

2.9.1 Ciphertext-only attack

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

Gesucht p_i, i= 1, ...,n oder k

2.9.2 known-plaintext attack

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

Gesucht k

2.9.3 chosen-plaintext attack

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

Gesucht k

3 Block-Cipher

Alphabet

Σ=$\{0,1\}$
$\Sigma^n := \Sigma \times \Sigma \times \cdots \times \Sigma$

Definition

Ein Block - Cipher ist eine **injektive** Abbildung
$C : K \to Perm(\Sigma^n)$
wobei K der Schlüsselraum ist.

Bsp.

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

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

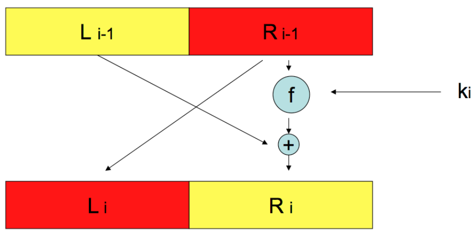
$\uparrow$Schlüssel

Frage:

Wie gross ist der Schlüsselraum K maximal?
$|K| \le (2^n)!$

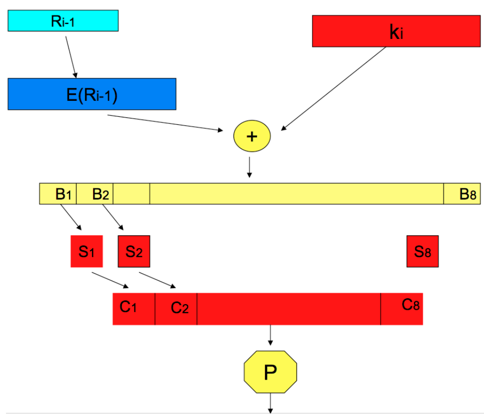
3.1 Data Encription Standard (DES)

Lucifer	Schlüssellänge	128
$\downarrow$		
DES	Schlüssellänge	56
	Blocklänge	64



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

Die f-Funktion:



3.2 Modi von Block-Cipher

Sei $\Sigma := \{0,1\}$
$p = c = \Sigma^4 = \{\square\square\square\square\}$
$k =$ Permutation von Σ^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)

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

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

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)

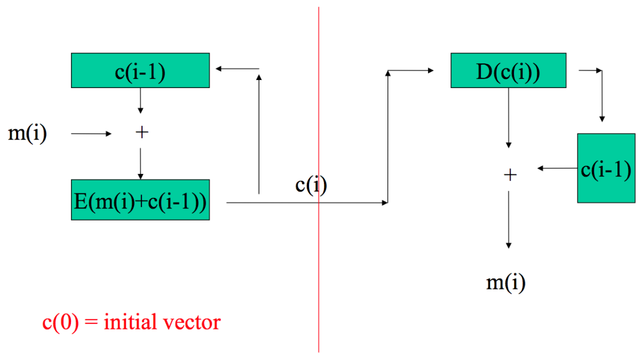
$m = \underset{\text{Länge n}}{m_1}|m_2|\ldots,\ n:$ Blocklänge

IV = Initialvektor (i.a. bekannt)
$C_0 := IV$
$C_1 := e_k(C_0 \oplus m_1)$
$C_2 := e_k(C_1 \oplus m_2)$

Bsp: $m = \underbrace{1100}_{m_1}|\underbrace{0110}_{m_2}|\underbrace{1100}_{m_3}|101$
$IV = C_0 = 1110$

$c_1 = e_k(c_0 \oplus m_1) = e_k(0010) = 0001$
$c_2 = e_k(c_1 \oplus m_2) = e_k(0111) = 1011$
$c_3 = e_k(c_2 \oplus m_3) = e_k(0111) = 1011$

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



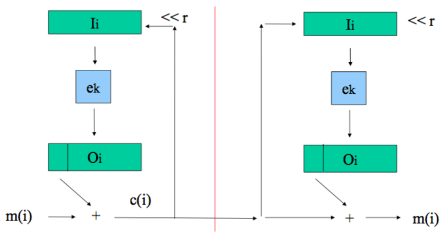
$m = \underset{\text{Länge n}}{m_1}|m_2,\ n:$ Blocklänge
$IV =$Initialvektor (i.a. bekannt)
$c_0 := IV,\ c_1 := e_k(c_0 \oplus m_1),\ c_2 := e_k(c_1 \oplus m_2)$
$c_1 \oplus d_k(c_2) = d_k(e_k(c_1 \oplus m_2)) = c_1 \oplus m_2 \oplus c_1 = m_2$
Bsp: $m = \underbrace{1100}_{m_1}|\underbrace{0110}_{m_2}|\underbrace{1100}_{m_3}|101,\ IV = c_0 = 1110$
$c_1 = e_k(c_0 \oplus m_1) = e_k(0010) = 0001$

$c_2 = e_k(c_1 \oplus m_2) = e_k(0111) = 1011$
$c_3 = e_k(c_2 \oplus m_3) = e_k(0111) = 1011$
Bem:

1. $m_1 = m_3 \not\Rightarrow c_1 = c_3$
2. Vertauschen kann bemerkt werden
3. Übertragungsfaktor machen sich bemerkbar

3.2.3 CFB-Modus (cipher feedback)

$m = \underbrace{\tilde{m_1}}_{\text{Länge}=r} |\tilde{m_2}|\tilde{m_3}|\ldots$, n: Cipher Block-Länge (DES: 64) und $\boxed{0 < r \leq n}$



$$
\begin{array}{ll}
I_1 & := IV \in \Sigma^n \\
\downarrow & \\
\boxed{\text{DES}} & \\
\downarrow & \\
O_1 : \underbrace{\square}_{r}\ \square & \\
\oplus & \rightarrow c_1 \\
\tilde{m_1} \rightarrow \underbrace{\square}_{r} &
\end{array}
$$

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

$$
\begin{array}{rcl}
I_1 = & 1110 & \\
& \downarrow & \\
& \boxed{e_k} & \\
& \downarrow & \\
O_1 & \mathbf{110}1 & \\
& \oplus & \rightarrow c_1 = 000 \\
\tilde{m_1} = & 110 &
\end{array}
\qquad
\begin{array}{rcl}
I_2 = & \overbrace{111\mathbf{0}}^{I_1}\overbrace{\mathbf{000}}^{c_1} & \\
& \downarrow & \\
& \boxed{e_k} & \\
& \downarrow & \\
O_2 & \mathbf{000}0 & \\
& \oplus & \rightarrow c_2 = 001 \\
\tilde{m_2} = & 001 &
\end{array}
$$

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

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

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

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

Beispiel:
$p = 11,\ q = 13$
$n = p * q = 143$
$\varphi(n) = 120 = 2^3 * 3 * 5$
e:=7 $\Rightarrow$ PK=(143,7)
$\mathbb{Z}_n = \{0, 1, 2, 3, \ldots, n-1\}$

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

$\Longrightarrow (*) \underbrace{e}_{7} *(-17) + 1 * \underbrace{\varphi(n))}_{120} = 1$ // mod $\varphi(n) \Rightarrow \boxed{d \equiv (-17) \text{ mod } \varphi(n)}$

4.2 Verschlüsselung und Entschlüsselung

4.2.1 RSA ist ein Blockcipher

encryption : enc
$\mathbb{Z}_n \longrightarrow \mathbb{Z}_n$
$m \longrightarrow c^e$ mod n

decyption : dec
$\mathbb{Z}_n \longrightarrow \mathbb{Z}_n$
$m \longrightarrow c^d$ mod n

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

4.2.2 Beweis

Fall 1:
ggT(m,n)=1
$(m^e)^d = m$ in $\mathbb{Z}_n$
Weil ggT(m,n)=1 existiert das Inverse von m: $\underbrace{m^{ed-1} = 1}_{\text{Das ist zu Zeigen!}}$ in $\mathbb{Z}_n$

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

Fall 2:
ggT(m,n)≠1 $\Rightarrow m = l * p$ oder $m = k * q$