Kryptographie

Fabio Oesch, Michael Künzli & Jan Fässler

4. Semester (FS 2013)

Inhaltsverzeichnis

1 Mathematische Grundlagen

1.1 Grundlagen

1.1.1 Einwegfunktion

In der Informatik ist eine Einwegfunktion eine mathematische Funktion, die komplexitätstheoretisch „leicht“ berechenbar, aber „schwer“ umzukehren ist.

1.1.2 Eigenschaften Funkionen

Injektiv Jedes x hat sein eigenes y.

Surjektiv Wenn das Bild der Wertemenge entspricht.

Bijektiv Wenn Injektiv wie Surjektiv

1.1.3 Kombinatorik

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

1.1.4 (erweiterter) Euklid

ggT(a,b):

$a = q * b + b_{neu}$

a	b	q	s	t
99	78	1		
78	21	3		
21	15	1		
15	6	2		
6	3	2		
3	0			

$s_1 = 1$ & $t_1 = 0$

a	b	q	s	t
99	78	1		
78	21	3		
21	15	1		
15	6	2		
6	3	2		
3	0		1	0

$s = t_{\text{alt}}$ & $t = s_{\text{alt}} - q \cdot t_{\text{alt}}$

a	b	q	s	t
99	78	1	-11	14
78	21	3	3	-11
21	15	1	-2	3
15	6	2	1	-2
6	3	2	0	1
3	0		1	0

Daraus folgt dann $3 = -11 \cdot 99 + 14 \cdot 78$

1.1.5 Diophantische Gleichungen

Die Gleichung $ax + by = c$ ist eine diophantische Gleichung. Sie besitzt nur eine ganzahlige Lösung, wenn $ggT(a, b) \| c$. Gibt es eine Lösung, gibt es unendlich viele Lösungen. Eine mögliche Lösung kann mit dem erweiterten euklidischen Algo. gefunden werden. Soll x oder y in einem bestimmten Bereich liegen, kann man sie mit folgender Formel shiften: $c = ax + by$, $0 = kab - kab$ addiert: $c = ax + kab + by - kab$, $c = a(x + kb) + b(y - ka)$

1.2 Modulares Rechnen

1.2.1 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: $23/27 \bmod 31$ // wenn ($ggT(27, 31) = 1 \rightarrow$ modulare Division möglich)
erweiterten euklidischen Algorithmus: $23 * \mathbf{7} * 31 + 23 * \mathbf{(-8)} * 27$ // 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: $27 * 2 \equiv 23 \mod 31$.

1.2.2 Potenzieren

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

1.) binäre Darstellung von b: $b = \sum_{i=0}^{k} \alpha_i 2^i$ mit $\alpha \in \{0, 1\}$.

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

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

Beispiel: $977^{2222} \mod 11$

1.) 2222_{10} ► bin $= 100010101110_2$

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

3.) Anwendung des Verfahren:

977	mod 11	$= 9$	5^2	mod 11	$= 3$
9^2	mod 11	$= 4$	3^2	mod 11	$= 9$
4^2	mod 11	$= 5$	$9 * 977$	mod 11	$= 4$
5^2	mod 11	$= 3$	4^2	mod 11	$= 5$
3^2	mod 11	$= 9$	$5 * 977$	mod 11	$= 1$
$9 * 977$	mod 11	$= 4$	1^2	mod 11	$= 1$
4^2	mod 11	$= 5$	$1 * 977$	mod 11	$= 9$
5^2	mod 11	$= 3$	9^2	mod 11	$= \mathbf{4}$
$3 * 977$	mod 11	$= 5$			

1.2.3 Chinesischer Restsatz

$x \equiv m_1 \mod n_1$
$x \equiv m_2 \mod n_2$
$x \equiv m_3 \mod n_3$

$N = n_1 \cdot n_2 \cdot n_3$, $N_i = \frac{N}{n_i}$

$ggT(N_i, n_i) = x \cdot n_i + y \cdot N_i = 1 \rightarrow e_i = y \cdot N_i$ // erweiterter Euklid

$x = m_1 \cdot e_1 + m_2 \cdot e_2 + m_3 \cdot e_3 \mod N$

1.3 Kettenbruch

$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots\, a_{n-1} + \cfrac{1}{a_n}}}}$. Für diesen Ausdruck schreiben wir auch: $< a_0; a_1, a_2, a_3, \dots, a_n >$

Berechnen mittels Euklid. Bsp: $\frac{37}{7} \rightarrow$ ggt$(37, 7)$
$37 = \mathbf{5} * 7 + 2$
$7 = \mathbf{3} * 2 + 1$
$2 = \mathbf{2} * 1 + 0 \rightarrow < 5; 3, 2 >$

Sei: $\alpha \in \mathbb{Q} \backslash \mathbb{Z}$ dann sei $< x_0; x_1, x_2, \dots, x_n >$ Approximation von α.

1.4 Gruppen

1.4.1 Notation

Eine Gruppe ist ein Tripel $(G, *, e)$. G ist eine Menge, $*$ eine Binärverknüpfung und e ein ausgezeichnetes Element der Menge G. Das Tripel muss folgende Eigenschaften erfüllen um eine Gruppe zu sein:

1. Assozivität: $\forall a, b, c \in G : f(f(a, b), c) = f(a, f(b, c))$
2. Neutrales Element: $\forall a \in G : f(a, e) = f(e, a) = a$
3. Inverses Element: $\forall a \in G : \exists b \in G : f(a, b) = e$

Die Abbildung f ist die Vorhinbezeichnete Abbildung $*$ zwischen den Mengen. Ist Gruppe zu dem noch Kommutativ, ist es eine abelsche Gruppe: $\forall a, b \in G : a * b = b * a$

1.4.2 Spezielle Mengen

Die Menge $\mathbb{Z}_n^*$ ist definiert als: $\{a \in \mathbb{Z}_n | ggT(a, n, =)1\}$.

1.4.3 Ordnung der Gruppe

Die Anzahl der Elemente in der Gruppe wird Ordnung genannt. Für eine Gruppe $\mathbb{Z}_n^*$ gilt $\phi(n)$

1.4.4 Erzeugende Elemente, zyklische Gruppen

Die Menge $< a >$ ist definiert als: $< a >:= \{an | n \in \mathbb{Z}\}$. Enthält die daraus erzeugte Menge aller Elemente der Gruppe, so wird die Gruppe als zyklisch bezeichnet. Zusätzlich wird definiert das $a^0 = e$ und $a^{-1} =$ invereses von a.

1.4.5 Gruppenhomomorphismus

Der Gruppenhomomorphismus ist eine strukturerhaltenden Abbildung. Seien $(G, \bullet, e_g)$ und $(H, \bullet, e_h)$. Das heisst, es gibt eine Abbildung f für die gilt: $f(x * y) = f(x) \bullet f(y)$. Das neutrale Element muss immer auf das neutrale Element der anderen Gruppe abgebildet werden!

1.4.6 Satz Euler Fermart

Der Satz lautet: $a^{\phi(n)} = 1 \bmod n$

1.5 Faktorisierung

1.5.1 Fermat

$n = ab$, n soll faktorisiert werden.

1. $t = \lfloor \sqrt{n} \rfloor + 1$.
2. Wenn $t^2 - n$ ein Quadrat weiter bei 4.
3. $t = t + 1$ Weiter bei 2.
4. $s := \sqrt{t^2 - n}, a = t + s, b = t - s$

Bemerkung: Der Alog terminiert immer, da gilt: $t \leq \frac{n+1}{2}$ Aufwand: $\frac{c^2}{8}$ und c ist eine geeignete Konstante. $\Delta := p - q < c\sqrt[4]{n}$ $\Rightarrow$ Resultat ist unabhängig von n!!!

1.5.2 Polard $p-1$

1. Alle Primzahlen q mit $q \leq B$
2. $\beta(q, B) = \max\{B \in \mathbb{N} : q^{\beta} \leq B\}$

Beispiel: $n = 1241143$, $B = 13$ Das B ist willkürlich gewählt. a wählen: $1 < a < n$ und $ggT(a, n) = 1$.
Menge bilden: $2, 3, 5, 7, 11, 13$ ($\forall$ $\mathbb{P}$ $<=$ B). Nun alle Exponenten suchen: $\beta(2, 13) = 3$, da $2^3 < 13$. Dann $\beta(3, 13) = 2)$, $\beta(5, 13) = \beta(7, 13) = \beta(11, 13) = \beta(13, 13) = 1$
$k = 2^3 * 3^2 * 5 * 7 * 11 * 13$ einsetzen in $ggT(a^k - 1, n) = 557 \in \mathbb{P}$

2 Definition eines Kryptosystems

2.1 Passiver Angreifer

Einer, der nur mithört.

2.2 Prinzip von Kerkhoffs

- Trennung von Verschlüsselungsmaschine und Schlüssel
- Die Verschlüsselungsmaschine ist dem Gegner bekannt
- Das zu schützende Geheimnis ist der Schlüssel

2.3 Formale Beschreibung Kryptosystems

System: (P, C, K, e, d), P Menge der Klartexte, C Menger der Geheimtexte, K Menge der Schlüssel: $e : K \times P \to C$; $d : K \times C \to P$ Es gilt: $\forall k \in K \forall p \in P : d(k, e(k, p)) = p \implies \forall k \in K : e(k, -)$ ist injektiv; $\forall k \in K : d(k, -)$ ist surjektiv.

2.4 Attacken auf Kryptosystem

1. **Ciphertext-onlyattack**
   Gegeben $c_i = ek(p_i)$, i = 1, ... , n
   Gesucht: p_i, i = 1, ... , n oder k
2. **Known-plaintextattack**
   Gegeben $(p_i, c_i = ek(p_i))$, i = 1, ... , n
   Gesucht: k
3. **Chosen-plaintextattack**
   Gegeben: $(p_i, c_i = ek(p_i))$, i = 1, ... , n
   pinach Wahl des Kryptoanalytikers
   Gesucht: k
4. **Chosen-ciphertextattack**
   Gegeben: $(c_i, p_i = dk(c_i))$, i = 1, ... , n
   c_i nach Wahl des Kryptoanalytikers
   Gesucht: k

3 Klassische Kryptographie

3.1 Klassen

Substitution Cipher	**Transposition Cipher**
Einheiten werden **ersetzt**.	Einheiten werden **vertauscht**.

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

$\Rightarrow \underbrace{\text{OUNO}}_{1}\underbrace{\text{EAUB}}_{2}\ldots$ **Bem.**

Einheiten werden vertauscht
(ABC ist Padding)

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

3.2 Caesar Cipher

Einfacher substitions Cipher bei welchem das Alphabet "verschoben" wird. Beispiel: A B . . . Z $\to$ B C . . . A. Sei $ord(c)$ eine Funktion, welche jedem Buchstaben einen numerischen Wert zuweise. Beispiel: $ord(a) = 0, ord(b) = 1$ etc. Weiter sei offset der Wert um welchen verschoeben werden soll. So kann mit $ord(\text{mc}) + offset \mod 26$ der Index des Buchstabens der Geheimnachricht bestummen berechnet. Auf gleiche Weise lässt sich ein Text entschlüsseln.

3.2.1 Attacken

Mit dem Wissen, in welcher Sprache die Nachricht geschrieben wurde, lässt sich mittels Häufigkeitsanalysen die Verschiebung bestimmen. Zum Beispiel ist das "e" in der deutschen Sprache ein sehr häufiger Buchstabe. Wird in der Geheimnachricht gleich häufig wie ein "e" zum Beispiel ein "i" entdeckt, kann die Verschiebung bestimmt werden.

3.3 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.

3.4 Playfair

3.4.1 Konstruktion Playfair-Quadrat

Keyword (Doppelte Streichen), dann der Rest des Alphabets ohne Buchstaben des Keywords.

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

3.4.2 Ver-/Entschlüsselung

Doppelbuchstaben durch X trennen	(PA AR ⇒ PA XA R)
Klartext:	LA BO UL AY EL AD YW IL LX
Geheimtext:	ME IK QO TX CQ TE ZX CO MW

Es werden jeweils Paare verschlüsselt.
(Entschlüsselung in entgegengesetzter Richtung der Verschlüsselung)
Wrap around and den Kanten.

Selben Zeile oder Spalte

$$\begin{array}{ccccc} * & E & * & * & * \\ * & C & * & * & * \\ * & L & * & * & * \\ * & Q & * & * & * \\ * & * & * & * & * \end{array}$$

$EL \rightarrow CQ$
1 nach unten

$$\begin{array}{ccccc} D & E & A & T & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array}$$

$AD \rightarrow TE$
1 nach rechts

Untersch. Zeilen bzw. Spalten

$$\begin{array}{ccccc} * & E & A & * & * \\ * & * & * & * & * \\ * & L & M & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array}$$

$LA \rightarrow ME$
Gleiche Zeile, Seite wechseln

3.5 Vigenère Cipher

Funktioniert gleich wie der caesar cipher, nur wird der Geheimtext nicht nur mit einer Verschiebung erzeugt sondern mit einem geheimen Satz (Schlüssel): Die Nachricht und der Schlüssel werden untereinander geschrieben. Dabei wird der Schlüssel solange wiederholt, bis unter jedem Buchstaben der Nachricht ein Buchstaben des Schlüssels steht. Die Buchstaben des Schlüssels geben die Verschiebung des Buchstabens der Nachricht an.

3.5.1 Kasiski Test

1. Paare von min. Länge 3 finden
2. Abstände innerhalb der Paare bestimmen
3. Abstände Faktorisieren
4. Häufigster Faktor ⇒ Kandidat für Schlüssellänge

3.5.2 Koinzidenz Index

$$\boxed{\mathbf{IC_{text}} = \sum_{i=1}^{n} p_i^2}$$

$$\boxed{\mathbf{IC_text} = \frac{\sum_{i=A}^{Z} n_i(n_i-1)}{N(N-1)}}$$

$IC_{eng} = 0.066895$
$IC_{ger} = 0.076667$
$IC_{flat} = 0.0385$

Gegeben: Alphabet A mit $|A| = 26$ $IC_{Lang} = \Sigma_{i=1}^{26} p_i^2$
Beispiel: Konsonanten gleiche Wahrscheinlichkeit p; I, O, U = doppelte Wahrscheinlichkeit $2p$; E,A = vierfache Wahrscheinlichkeit $4p$. Zusammen zählen: $21p+3*2p+2*4p = 35p$. Für die Formel gilt dann: $21\frac{1}{35}^2+3\frac{2}{35}^2+2\frac{4}{35}^2$

3.5.3 Berechnung der Schlüssellänge

Gegeben: C: Vigenère-Chiffrat der Länge n mit der Schlüssellänge p. $\gamma = \alpha IC_L + \beta IC_{flat}$ α:Anzahl Buchstabenpaare aus der selben Spalte. β Anzahl Buchstabenpaare aus verschiedenen Spalten. γ Anzahl gleicher Buchstaben aus C. IC: Koinzidenzindex des Chiffrats, IC_L Koinzidenzindex der Sprache.
Formel: $p = \frac{n*(IC_L - IC_{Flat})}{IC(n-1)+IC_L - n*IC_{Flat}}$

3.6 Vigenères Chipres

3.6.1 Beschreibung

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

3.6.2 Kryptoanalysis des Vigenère-Cipher

1) Schlüssellänge p=1,2,3,...

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

2) Sei s,t zwei Strings über dem Alphabet A: $s = s_1, s_2, s_3, \ldots .s_k$ / $t = t_1, t_2, t_3, \ldots, t_l$
Seien $n_1(s) :=$ #A's in s, $n_2(s) :=$ #B's in s, ...

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

Beispiel: s="AABCCA"/ t="ABCABCABC"

$$\left.\begin{array}{l} n_1(s) = 3, n_1(t) = 3 \\ n_2(s) = 1, n_2(t) = 3 \\ n_3(s) = 2, n_3(t) = 3 \end{array}\right\} \rightarrow MIC(s,t) = \frac{1}{6*9}[3*3 + 1*3 + 2*3]$$

3.) Anwendung auf Cipher Text

$(i,j)\backslash k$	0	1	2	...
$(1,2)$				
$(1,3)$				
$(1,4)$				
$(1,5)$				
$(2,3)$			$MIC(c_2, c_{3+2})$	
$(2,4)$				
$(2,5)$				
$(3,4)$				
$(3,5)$				
$(4,5)$				

p = Schlüssellänge von c (Annahme:5)
$c_1, c_2, ..., c_5$ Abschnitte des Ciphertext
$i = 1, \ldots, p$
$j = i+1, \ldots, p$
$k = 0, \ldots, 25$
$\rightarrow MIC(c_i, c_{j+k})$

Beispiel:
c_1:AXBM...
c_3:ABXH...
$------$
c_{3+2}:CDZJ...

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. $klar_1 \sim klar_2$

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

Wir suchen die grossen Werte von $MIC(c_i, c_j + k)$
$MIC(c_i, c_j + k)$ gross $\iff c_i \sim c_j + k$

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

↓ 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: $\beta_1 = 0, 1, \ldots, 25$

3.7 One-Time-Pad

$$\Sigma = \{0,1\} \quad \begin{array}{ll|c|l} \text{Klartext:} & p_1p_2p_3p_4p_5\cdots = & 0 & 0101\ldots \\ \text{Schlüssel:} & k_1k_2k_3k_4k_5\cdots = & 1 & 0110\ldots \\ \text{ciphertext:} & \underset{p_1\oplus k_1}{c_1}\ c_2c_3c_4c_5\cdots = & 1 & 0011\ldots \end{array}$$

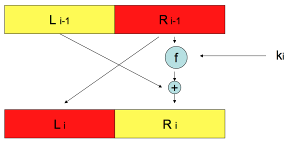
One-Time-Pad gilt als sicher, da es zu einem Ciphertext mehr als nur eine verständliche Nachricht gibt. Der Angreifer kann nicht entscheiden, welcher mögliche Schlüssel nun der richtige ist, vorausgesetzt, der Schlüssel ist solange wie die Nachricht selbst und wurde zufällig erzeugt.

4 Block-Cipher

Definition Block-Cipher: $C : K \rightarrow Perm(\Sigma^n)$. K ist der Schlüsselraum. Σ = Alphabet.
Maximale Grösse Schlüsselraum: $|\Sigma|^n!$ wobei n = Schlüssellänge.

4.1 Data Encription Standard (DES)

Die Schlüssellänge von DES beträgt 56.



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

4.2 Betriebsmodi

4.2.1 ECB-Modus (electronic code block)

Abarbeiten der Blöcke ohne spezielles Verfahren // **Bem:** $m_1 = m_3 \Rightarrow c_1 = c_3$

$m = \underbrace{1100}_{m_1}|\underbrace{0110}_{m_2}|\underbrace{1100}_{m_3}|101^* \xrightarrow[m_1]{} \boxed{e_k} \xrightarrow[c_1]{}$

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

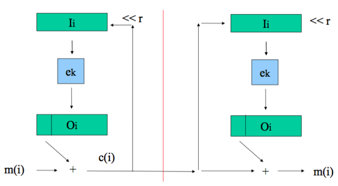
$c_1 = e_k(c_0 \oplus m_1) = e_k(0010) = 0001$
$c_2 = e_k(c_1 \oplus m_2) = e_k(0111) = 1011$
$c_3 = e_k(c_2 \oplus m_3) = e_k(0111) = 1011$

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)
$c_0 := IV$, $c_1 := e_k(c_0 \oplus m_1)$, $c_2 := e_k(c_1 \oplus m_2)$
$c_1 \oplus d_k(c_2) = d_k(e_k(c_1 \oplus m_2)) = c_1 \oplus m_2 \oplus c_1 = m_2$
Bem: $m_1 = m_3 \not\Rightarrow c_1 = c_3$
Bem: Wenn ein C-Block korrupt ist, ist der M-Block und der danach folgende auch korrupt, alle anderen aber wieder richtig.

4.2.3 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 $\boxed{0 < r \leq n}$

Modus



Beispiel

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

$$\begin{array}{ccc|ccc}
I_1 = & 1110 & & I_2 = & \overbrace{111\mathbf{0}}^{I_1}\overbrace{\mathbf{000}}^{c_1} & \\
& \downarrow & & & \downarrow & \\
& \boxed{e_k} & & & \boxed{e_k} & \\
& \downarrow & & & \downarrow & \\
O_1 & \mathbf{110}1 & & O_2 & \mathbf{000}0 & \\
& \oplus & \to c_1 & & \oplus & \to c_2 \\
\tilde{m}_1 = & 110 & & \tilde{m}_2 = & 001 &
\end{array}$$

5 RSA

5.1 Schlüsselerzeugung

PK = (n,e) und SK = (n,d)

1. Wähle $p, q \in \mathbb{P} : p \neq q$
2. $n = pq$
3. Wähle $e \in \mathbb{N}^* : 1 < e < \phi(n)$ und $ggT(e, \phi(n)) = 1$ // $\phi(n) = (p-1)(q-1) = |\mathbb{Z}_n^*|$
4. Finde: $d \in \mathbb{N}^* : 1 < d < \phi(n)$ und $ggT(d, \phi(n)) = 1$ // $d := e^{-1}$ in $\mathbb{Z}_{\varphi(n)}^*$

5.2 Verschlüsselung und Entschlüsselung

encryption : $m \longrightarrow m^e$ mod n

decyption : $m \longrightarrow c^d$ mod n

5.3 Attacken

5.3.1 3 Primzahlen für zwei Schlüssel

Der RSA Provider verwendet p_1, q_1, p_2 um zwei Schlüssel zu erzeugen. Es gilt: $n_a = p_1 q_1$ $n_b = p_2 * q_1$ Dies impliziert: $ggt(n_a, n_b) = q_1$

5.3.2 Gleicher Modul

h berechnen: $h = \frac{e_b d_b - 1}{ggt(e_a, e_b d_b - 1)}$. Der Angreiffer muss nun nur noch folgende Gleichung lösen: $\alpha h + \beta e_a = 1$ β ist die Zahl, mit welcher der Angreifer die Nachrichten von Alice entschlüsseln kann. Ist $\beta < 0$ dann muss $+h$ gerechnet werden.

5.3.3 Hastad Attack

$$
\begin{array}{lllll}
 & & \color{red}{e = 3} & & = x \\
 & & \text{Bob } (n_1, e) & : \color{red}{c_1} & = m^3 \text{ mod } n_1 \\
 & \nearrow & & & \\
\text{Alice} & \rightarrow & \text{Jon } (n_2, e) & : \color{red}{c_2} & = m^3 \text{ mod } n_2 \\
 & \searrow & & & \\
 & & \text{Paul } (n_3, e) & : \color{red}{c_3} & = m^3 \text{ mod } n_3
\end{array}
$$

Chinesischer Restsatz: $m^3 = crt([m^3, m^3, m^3], [n_1, n_2, n_3])$
Es benötigt so viele Gleichungen für den Restsatz wie e gross ist.

5.3.4 Wiener's Angriff

Voraussetzung

- $(n, e), (n, d)$ RSA-Schlüssel mit $n = p \cdot q$, $p < q < 2p$
- $0 < d \leqslant \frac{1}{3} \sqrt[4]{n}$

Behauptung $\exists$ schneller Alg. zur Faktorisierung von n

Konvergenten von $\frac{e}{n} = \frac{K}{\delta}$ $\phi = \frac{e\delta_i - 1}{K_i}$
Bsp: $(n, e) = (2021, 773)$, $(n, d) = (2021, ?)$

Kettenbruch von $\frac{e}{n}$

$$\begin{aligned}773 &= 0 * 2021 + 773\\ 2021 &= 2 * 773 + 475\\ 773 &= 1 * 475 + 298\\ 475 &= 1 * 298 + 177\\ 298 &= 1 * 177 + 121\end{aligned}$$

Konvergenten: $0, \frac{1}{2}, \frac{1}{3}, \frac{2}{5}, \frac{4}{7}, \ldots$

$$\begin{aligned}\phi_1 &= \frac{e * 2 - 1}{1} = 1545\\ \phi_2 &= \frac{e * 3 - 1}{1} = 2318\\ \phi_3 &= \frac{e * 5 - 1}{1} = 1932\\ \phi_4 &= \frac{e * 7 - 1}{4} = 1352.5 \notin \mathbb{Z}\end{aligned}$$

$\frac{5}{3} =$

$$\begin{array}{r|c|l} 5 = & 1 & \cdot 3 + 2\\ 3 = & 1 & \cdot 2 + 1\\ 2 = & 2 & \cdot 1 \end{array}$$

KE von $\frac{5}{3} =< 1, 1, 2 >$

$1 + \frac{1}{1+\frac{1}{2}} = \frac{5}{3}$

Voraussetzung

- $(n, e), (n, d)$ RSA-Schlüssel mit $n = p \cdot q$, $p < q < 2p$
- $0 < d \leqslant \frac{1}{3}\sqrt[4]{n}$

Behauptung $\exists$ schneller Alg. zur Faktorisierung von n

```
e = d.invers_mod(phi)
con = continued_fraction_list(e/n, partial_convergents = true)
```

5.3.5 de Wenger - Spezialfall: RSA-Schlüssel

Vor.

1. $n = p * q$, $p, q \in \mathbb{P}^*$, $p > q$
2. $\delta = p - q$

Beh. $0 < p + q - 2 * \sqrt{n} < \frac{\delta^2}{4*\sqrt{n}}$

Bew.

1. $\delta^2 = (p-q)^2 = (p+q)^2 - 4 * n = (p + q - 2 * \sqrt{n})(p + q + 2 * \sqrt{n}) > 0$
2. $p + q - 2 * \sqrt{n} = \frac{\delta^2}{(p+q)+2*\sqrt{n}} < \frac{\delta^2}{2*\sqrt{n}+2*\sqrt{n}} = \frac{\delta^2}{4*\sqrt{n}}$

5.3.6 SK-Such-Algorithmus

Gegeben: Bob: (n, e), (n, d) und Alice: (n, d_A)
unbekannt: $d_A, p, q, \varphi(n)$ Finde d_A
Bemerkung:
Falls $c = m^{m_A} \bmod n$ ist, gilt $m = c^{d_A} \bmod n$
$e * d - 1 = k * \varphi(n)$
$h := \frac{e*d-1}{ggT(e_A, e*d-1)}$ $\qquad //\varphi(n) \mid h$
$h := \frac{h}{ggT(h, e_A)}$
$e_A \cdot \alpha + h \cdot \beta = 1 \Rightarrow d_A := \alpha \bmod h$

6 Signaturen

6.1 RSA

Alice: $(m, h(m)) \mapsto (m, sig(h(m)))$. $d(sig_{Alice}, h(m)$
Alice signiert mit: $h(m)^d_{\text{Alice}} \mod n = s$
Der Empfänger prüft: $s^e \mod n$

6.2 Das Lamport-Schema

Gegeben: Ein-weg-Funktion: $f : Y \to Z$
$m = x = (x_1, x_2, \dots, x_n)$ mit $x_i \in \{0, 1\}$
Jedes Bit wird einzeln signiert!

1. Wähle **zufällig** Elemente aus Y
   $\left.\boxed{\begin{matrix} y_{10} & y_{11} \\ y_{20} & y_{21} \\ \vdots & \vdots \\ y_{k0} & y_{k1} \end{matrix}}\right\}$**Geheim**
2. Berechne und publiziere: $f(y_{j,j}) =: z_{j,j}$
   $\boxed{\begin{matrix} z_{10} & z_{11} \\ z_{20} & z_{21} \\ \vdots & \vdots \\ z_{k0} & z_{k1} \end{matrix}}$
3. Signieren von x_i: $sig(x_i) = \begin{cases} y_{i0}, \text{ falls } x_i = 0 \\ y_{i1}, \text{ falls } x_i = 1 \end{cases}$

Bsp: $x = (0, 0, 1)$ $\boxed{\begin{matrix} z_{10} & z_{11} \\ z_{20} & z_{21} \\ z_{30} & z_{31} \end{matrix}}$ $sig(X) = y_{10},\ y_{20},\ y_{31} \xrightarrow{f} f(y_{10})$

6.3 El-Gamal

Ein Signaturverfahren besteht aus 4 Mengen: P Nachrichtenmenge, S Signaturmenge, SK der geheime Schlüssel, PK der öffentliche Schlüssel. Dazu gibt es das Signaturverfahren:

$$sig : P \times SK \to S \tag{1}$$
$$(m, k) \mapsto sig(m, k) \tag{2}$$

und ein Verifikationsverfahren: $ver : P \times S \times SK \to 0, 1$
Vorgehen:

- Man wähle ein grosses $p \in \mathbb{P}$
- Man wähle ein Erzeugens: $\omega \in \mathbb{Z}_p^*$. p und ω sind öffentlich bekannt.
- Jeder Teilnehmer (T) wählt zufällig $a_T < p - 1 \mod p$. Diese Zahl ist der geheime Schlüssel!
- T berechnet $b_T = \omega^{a_T}$ und veröffentlicht diese Zahl.
- T wählt zufällig $r \in \mathbb{Z}_{p-1}^*$

Die Mengen sind dann wie folgt definiert:

- $P = \mathbb{Z}_p^*$, $S = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$
- Einer Nachricht $m \in \mathbb{Z}_p^*$ wird ein Paar $(x, y) \in \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ zugeordnet.

- T berechnet: $x = \omega^r \mod p$
- T berechnet: $y = (m - a_T x) r^{-1} \mod p-1$.
- Das Tripel (m,x,y) ist das signierte Dokument. Verifikation: $b_T^x x^y \equiv \omega^m \mod p$
- $r^{-1} = rx \equiv 1 \mod p-1$

Bsp: $p = 41$
$\omega := 7 \in Gen_p$
$m = 13$ zu signieren:

$$\begin{aligned} a_t &:= 5 \\ b_t &:= \omega^{a_T} \mod p = 7^5 \mod 41 = 38 \end{aligned}$$

Wähle $r \in \mathbb{Z}_{p-1}^* = \mathbb{Z}_{40}^*$: Sei $r = 3$, $r^{-1} = 27 \mod 40$

$$\begin{aligned} x &= \omega^r \mod p = 7^3 \mod 41 = 15 \\ y &= (m - a_T \cdot x) \cdot r^{-1} \mod p-1 = (13 - 5 \cdot 15)27 \mod 40 = 6 \end{aligned}$$

$\Rightarrow (m, sig(m)) = (13, 15, 6)$

6.3.1 Verifikation

Signatur $sig(m) = (m, x, y)$ ist gültig $\Leftrightarrow$ (*) $\boxed{b_T^x \cdot x^y \equiv \omega^m \mod p}$

Bem: $y \equiv (m - a_T \cdot x) \cdot r^{-1} \mod p-1 \Leftrightarrow a_T \cdot x + r \cdot y \equiv m \mod p-1 \Leftrightarrow \exists k \in \mathbb{Z} : a_T \cdot x + r \cdot = m + k(p-1)$
$\alpha \equiv \beta \mod n \Leftrightarrow n \mid \alpha - \beta$

Bew: $\Rightarrow$ $$\underbrace{\underbrace{b_T^x}_{(\omega^{a_T})^x} \cdot \underbrace{x^y}_{\cdot(\omega^r)^y}}_{\omega^{a_T \cdot x + r \cdot y} = \omega^{m+k(p-1)} = \omega^m \cdot (\omega^{p-1})^k \overset{\text{Fermat}}{\equiv} \omega^m \cdot 1 \mod p} \equiv \omega^m \mod p$$

ω Erzeugendes von $\mathbb{Z}_p^*$: $\omega^i \equiv \omega^j \mod p$ und $0 < i, j < p-1 \Rightarrow i = j$

7 Hashfunktionen

$h : \Sigma^* \to \Sigma^n$

One-way Funktion: Gegeben $y \in \Sigma^n$ Gesucht $x \in \Sigma^* : h(x) = y$

Weak collision free: nGegeben: $x_1 \in \Sigma^*$ Gesucht: $x_2 \in \Sigma^*$ mit $x_1 \neq x_2$ und $h(x_1) = h(x_2)$

Strong collision free: Gesucht: $x_1, x_2 \in \Sigma^*, x_1 \neq x_2$ mit $h(x_1) = h(x_2)$ Eigenschaften gegeben, wenn Problem(e) nicht lösbar.

Das One-Way Problem ist das schwiriegste Problem von allen dreien, daher ist eine Funktion 'relativ schnell eine One-Way-Function'.

7.1 Angriff

Wenn $\sqrt{n}$ in nützlicher Frist berechnet werden kann. n Bezeichnet die Grösse der Menge der möglichen Hashes. Bekannt unter 'Birthday Attack'.

7.2 SHA-1

Die Länge des Hash-Strings ist immer 160 Zeichen. Internt arbeitet SHA-1 mit 512er Blöcken. Bsp Block:

$$\overbrace{\underbrace{\boxed{0}\boxed{\qquad\qquad}}_{448}\underbrace{\boxed{\qquad}\boxed{511}}_{64}}^{512}$$

$\underbrace{0110011011001010}_{m}10\ldots010000$, $|m| = 16 = (10000)_2$

8 Kryptographische Protokolle

8.1 Needham-Schroeder

Ziel: Nur eine Instanz, die Kenntnis von $K_{A,TTP}$ hat, kann isch als Alice ausgeben
Zweck: Alice Bob möchte sich authentifizieren.

TTP:= Trusted Third Party
r := Randomzahl
$\{\}_K$:= Inhalt verschlüsselt mit Schlüssel K

1. $A \to TTP : (A, B, r_A)$
2. $TTP \to A : \{r_A, B, K_{A,B}, \{K_{A,B}, A\}_{K_{B,TTP}}\}_{K_{A,TTP}}$
3. $A \to B : \{K_{A,B}, A\}_{K_{B,TTP}}$
4. $B \to A : \{r_B\}_{K_{A,B}}$
5. $A \to B : \{r_B + 1\}_{K_{A,B}}$

8.2 Diffie-Hellman-Protokoll

Alice$\overset{?}{\longleftrightarrow}$Bob
Gegeben: $\boxed{p \in \mathbb{P}^* \text{ (gross)}, \omega \in Gen_P \text{ Erzeugendes von } \mathbb{Z}_p^*}$ bekannt

	Alice	Bob	
Wähle	α	β	zufällig $1 < \alpha, \beta < p - 1$
Berechne	$a = \omega^\alpha \mod p$	$b = \omega^\beta \mod p$	
Sende	$\to$Bob: a	$\to$Alice: b	öffentlich bekannt!
Berechne	$k_a = b^\alpha \mod p$	$k_b = a^\beta \mod p$	

Behauptung: $k := k_a = k_b$
Beweis: $k_a = b^\alpha = (\omega^\beta)^\alpha = \omega^{\beta\cdot\alpha} = \omega^{\alpha\cdot\beta} = (\omega^\alpha)^\beta = a^\beta = k_\beta$ in $\mathbb{Z}_p^*$
Aufgabe: Gegeben: (p, ω, a, b)

```
p = nth_prime(2000) = 17389
w = 2
a = 1000 = w^alpha = 2^alpha mod p => k = b^alpha mod p
b = 500
```