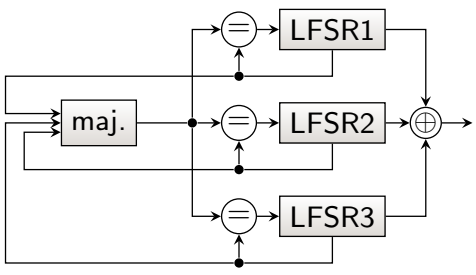
10 Stromchiffren

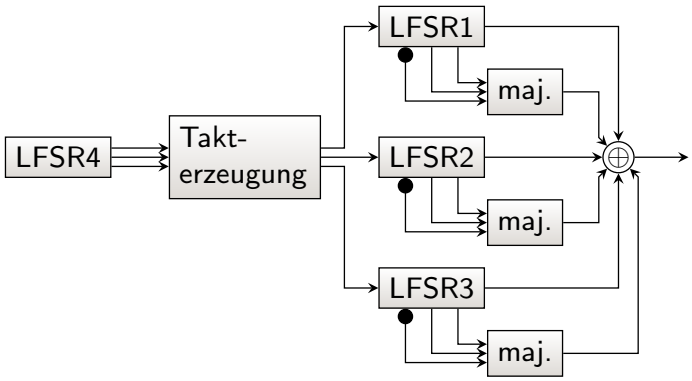
- Blockchiffren im Output-Feedback-Modus (OFB) dienen lediglich dazu, einen quasi-nichtperiodischen Schlüssel zu erzeugen, der mit dem Klartext XOR-verknüpft wird
- Verfahren, die von vornherein nur für Erzeugung eines quasi-nichtperiodischen Schlüssels ausgelegt sind, heißen Stromchiffren
- einfachster Ansatz: linear rückgekoppeltes Schieberegister (LFSR) – aber anfällig gegen Klartext-Geheimtext-Kompromittierung (siehe 3.12)
- viele Ansätze modifizieren LFSRs oder kombinieren mehrere, um Sicherheit zu erhöhen
- wichtige Kenngröße: lineare Komplexität – wie lang wäre ein LFSR mindestens, das die gleiche Bitfolge erzeugt?

10.1 A5/1 und A5/2

- zur Verschlüselung zwischen Mobiltelefon und Basisstation in GSM-Netzen verwendet
- drei LFSRs mit den Längen 19, 22, 23
- Ausgabe als Verknüpfung der Ausgabe der drei LFSR mit fester (bekannter) Rückkopplung
- Schlüssel und Frame-Nummer definieren die initiale Belegung (64 bit)
- in A5/1 (stärkere Variante) Ausgabe-Verknüpfung XOR, aber Taktung der LFSR von Inhalt abhängig:
  - betrachte jeweils eine mittlere Position in jedem Register
  - Register wird getaktet, wenn mittleres Bit gleich des Mehrheitswerts der drei mittleren Bits



- in A5/2 (schwächere Variante) kompliziertere Ausgabe-Verknüpfung, aber Taktung durch viertes LFSR gesteuert
  - Ausgabe XOR-Verknüpfung der einzelnen Registerausgaben und der Merhheitswerte ausgewählter mittlerer Positionen, wovon einer negiert
  - Taktungslogik ähnlich wie in A5/1 (Vergleich mit Mehrheitswert), aber aus Mittelabgriffen eines vierten LFSR (Länge 17) gespeist
  - LFSR1–LFSR3 ist je ein Bit in LFSR4 zugeordnet; Taktung, wenn dieses Bit gleich dem Mehrheitswert aller drei



- A5/2 ist bei Klartext-Geheimtext-Kompromittierung (Bitstrom, mit dem XOR-verknüpft wird, bekannt) zu brechen
  - Annahme des Anfangszustands von LFSR4, sodass Taktung bekannt
  - bei Initialisierung von A5/2 wird in jedem LFSR ein bestimmtes Bit gesetzt
  - da Majoritätsfunktion darstellbar als quadratische Funktion $(\mathrm{maj}(a,b,c) = a \cdot b \oplus b \cdot c \oplus c \cdot a$, Ausgabe lineare Funktion der $18 + 21 + 22 = 61$ unbekannten initalen Bits und der $\frac{18 \cdot 17}{2} + \frac{21 \cdot 20}{2} + \frac{22 \cdot 21}{2} = 594$ unbekannten paarweisen Produkte
  - bekannte Ausgabe liefert lineares Gleichungssystem mit 655 Unbekannten, das nach der Initialbelegung aufgelöst werden kann
  - alle $2^{16} = 65\,536$ möglichen Belegungen von LFSR4 durchprobieren (vorberechnete und tabellierte Zwischenergebnisse erlauben viele davon frühzeitig auszuschließen)
- GSM liefert hinreichend Kompromittierung, die ähnlich zu Klartext-Geheimtext-Kompromittierung ist, da Verschlüsselung nach Fehlerschutz angewendet wird
  - der Fehlerschutz ist derart, dass es eine Rechteckmatrix $\boldsymbol{H}$ gibt, sodass für einen Klartextblock (Frame) $\boldsymbol{x}$ gilt: $\boldsymbol{xH} = \boldsymbol{0}$
  - für den verschlüsselten Frame $\boldsymbol{y} = \boldsymbol{x} \oplus \boldsymbol{k}$ gilt daher $\boldsymbol{yH} = \boldsymbol{xH} \oplus \boldsymbol{kH} = \boldsymbol{kH}$
  - da $\boldsymbol{y}$ und $\boldsymbol{H}$ bekannt sind, ergibt sich ein System linearer Gleichungen, die der Schlüssel-Bitstrom $\boldsymbol{k}$ erfüllen muss
  - Einsetzen der linearen Gleichungen für die Ausgabe (für eine bestimmte Belegung von LFSR4) gibt wiederum ein lineares Gleichungssystem, das für die Initialbelegung der anderen LFSRs aufgelöst werden kann

- dieselbe Beinahe-Klartext-Geheimtext-Kompromittierung erlaubt auch Angriff auf A5/1
- Details komplizierter und Aufwand größer, da Taktung vom kompletten Initialzustand abhängt
- Barkan, Biham und Keller zeigten 2003, dass aus 20 min GSM-Daten in Echtzeit der Schlüssel berechnet werden kann, wenn man auf 600 GB vorberechnete Tabellen zurückgreift

10.2 RC4

- ähnlich zu Blockchiffre im Output-Feedback-Modus (OFB) mit 8 bit-Blöcken
- interner Zustand: zwei 8 bit-Werte i, j und eine Tabelle S mit $2^8 = 256$ Einträgen zu je 8 bit (S-Box)
- Erzeugung Byte des Schlüsselstroms:
  1. $i \leftarrow i + 1 \mod 256$
  2. $j \leftarrow j + S[i] \mod 256$
  3. vertausche i-ten und j-ten Eintrag in S
  4. Ausgabe $K = S[S[i] + S[j]]$
- sehr einfach und effizient implementierbar – deutlich schneller als DES
- Schlüssel wird zur Initialisierung von S verwendet:
  1. setze $S[0] = 0, S[1] = 1, \ldots S[255] = 255$
  2. setze $K[0], \ldots, K[255]$ auf (periodisch wiederholten) Schlüssel
  3. wiederhole für $i = 0, \ldots, 255$:
     - 3.1 $j \leftarrow j + S[i] + K[i] \mod 256$
     - 3.2 vertausche i-ten und j-ten Eintrag in S

- RC4 wird bei WEP (alter WLAN-Sicherheitsstandard) mit einem 24 bit-Initialisierungsverktor verwendet: die ersten 3 Byte des Schlüssels sind der in jedem Packet andere, im Klartext übertragene IV, die übrigen ursprünglich 5, später bis zu 29 Byte sind der geheime Schlüssel
- wird ein Packet mit dem IV $\begin{pmatrix}3 & 255 & X\end{pmatrix}, X < 250$ aufgefangen, so ergeben sich folgende erste Runden bei der Bestimmung der S-Box:

$S[0]$	$S[1]$	$S[2]$	$S[3]$	$\cdots$
0	1	2	3	$\cdots$
				$i = 0,\ j \equiv 0 + S[0] + K[0] \equiv 3$
3	1	2	0	$\cdots$
				$i = 1,\ j \equiv 3 + S[1] + K[1] \equiv 3$
3	0	2	1	$\cdots$
				$i = 2,\ j \equiv 3 + S[2] + K[2] \equiv 5 + X$
3	0	5+X	1	$\cdots$
				$i = 3,\ j \equiv 5 + X + S[3] + K[3] \equiv 6 + X + K[3]$
3	0	5+X	6+X+K[3]	$\cdots$

  Annahme: $6 + X + K[3] \geq 4$
- mit einer Wahrscheinlichkeit von etwa 5 % ändern sich diese Einträge in den übrigen Runden nicht mehr

- gilt $S[0] = 3, S[1] = 0, S[3] = 6 + X + K[3]$, so lautet die erste Ausgabe des RC4-Algorithmus:
  1. $i \leftarrow 0 + 1 \equiv 1 \mod 256$
  2. $j \leftarrow 0 + S[1] \equiv 0 \mod 256$
  3. vertausche ersten und nullten Eintrag in S
  4. Ausgabe $K = S[S[0] + S[1]] = S[0 + 3] = 6 + X + K[3]$
- durch den Aufbau der Nutzdaten erstes Byte gut vorhersagbar $\Rightarrow$ K aus Geheimtext ableitbar $\Rightarrow$ erstes Byte des geheimen Schlüssels $K[3]$ bestimmbar, da X bekannt
- da nur etwa 5 % der passenden IV zum richtigen Ergebnis führen, etwa 60 Packets mit passendem IV nötig, um $K[3]$ relativ sicher zu bestimmen
- Ansatz iterierbar, um sukzessive den gesamten Schlüssel aufzudecken (Aufwand steigt nur linear mit Schlüssellänge!)
- in aktiven WLANs in relativ kurzer Zeit genügend Packets mit passenden IVs
- in WPA-TKIP (WEP-Nachfolger) wird noch RC4 verwendet, aber IV und geheimer Schlüssel werden auf kompliziertere Weise verknüpft, um obigen Angriff zu vereiteln
- WPA2 wechselt zu AES
- grundsätzliches Problem von RC4: je nach Position bestimmte Zeichen im Schlüsselstrom etwas wahrscheinlicher als andere