Übertragungstechnik I und II

4. Sicherheitskodierung und -Dekodierung
   - 4.1 zur Einordnung
   - 4.2 Ziele
   - 4.3 Mittel der Kryptographie
   - 4.4 einige Verfahren

4 Sicherheitskodierung und -Dekodierung
4.1 zur Einordnung (1)

- Überschneidungen mit der Kanalkodierung
  → deshalb beides hier kurz vorgestellt:

<u>Sicherheitskodierung</u>

- Schutz vor
  - ..... Beeinflussung der Nachricht
  - (..... Beeinflussung der Nachricht)
  - ..... Kenntnisnahme des Inhalts der Nachricht

<u>Kanalkodierung</u>

- Schutz vor
  - (..... Beeinflussung der Nachricht)
  - ..... Beeinflussung der Nachricht

- Informationstheorie -
- Mathematik -
- Kryptographie -

kommt mehr aus der Datenwelt

kommt mehr von der Untersuchung von Kanälen

4.1 zur Einordnung (2)

- Die Sicherheitskodierung befindet sich in der Regel nach der Quellkodierung.
- In manchen Fällen leistet die Sicherheitskodierung schon das mit, was für die Übertragung des Signals sonst in der Kanalkodierung realisiert wird (z.B. .....).
- In der Regel werden Sicherheitskodierung und Kanalkodierung nacheinander angewendet.
- Da die Kanalkodierung mehr Bezug zur Physik des Kanals hat, befindet sie sich dann dichter am Kanal.
- Im Bezug zu der jeweiligen konkreten Anwendung der Datenübertragung kann die Sicherheitskodierung an verschiedenen Stufen der Verarbeitung stehen (vergleiche OSI-Modell, das aber hier nicht angewendet wird).
  Die Sicherheitskodierung kann auf hoher Ebene, dicht an der Anwendung, erfolgen (z. B. Mail-Programm) oder aber auch dicht am Übertragungskanal (z. B. IP-VPN).

4.2 Ziele (1)

- Schutz der Datenvertraulichkeit (.....)
  - .....

    .....
- Schutz der Datenintegrität (.....)
  - .....
- Sicherung der Authentizität (.....)
  - .....

4.3 Mittel der Kryptographie (1)

- Verschlüsselung - zuerst: Angriffsstrategien
  - **Ciphertext-Only-Angriff**
    - nichts bekannt außer .....
    - Ziel sind ..... und **.....**
  - **Known-Plaintext-Angriff**
    - verschlüsselte Nachricht und Klar-“text“ oder Teil davon bekannt
    - Brute-Force-Angriff
      - z. B. DES, 56 bit-Schlüssel → 72.057.594.037.927.936 Varianten
        falls 1 µs je Variante → 1.142 Jahre (Treffer bei 50% aller Fälle)
        Jahr 2000: NSA braucht vermutlich 1 Sekunde (Stand ca. 2007)
    - Ziel ist **.....**
  - **Chosen-Plaintext-Angriff**
    - Einspeisen von Klartext und Analyse Ciphertext
    - Ziel ist Schlüssel und / oder Verfahren

4.3 Mittel der Kryptographie (2)

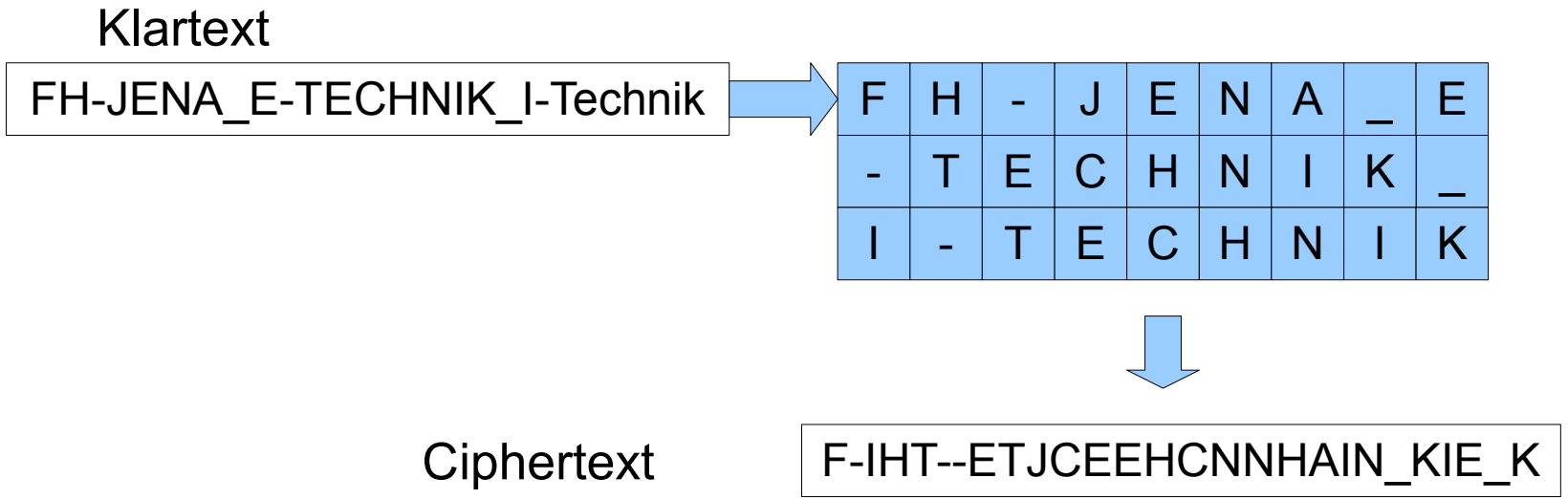
- Verschlüsselung
  - Substitution, monoalphabetisch
    - Symbol N am Eingang → Symbol M am Ausgang, N,M .....
    - eineindeutige Zuordnung der Symbole
    - sehr einfach und alt
    - sehr schwach
    - Analyse Ciphertext allein ist sehr erfolgversprechend (Prinzip)
  - Substitution, polyalphabetisch (z. B. Enigma)
    - Symbol N am Eingang → Symbol M am Ausgang, N,M .....
    - Verwischen von Symbolhäufigkeiten
    - eindeutige Zuordnung der Symbole bei der Dekodierung
    - relativ einfach (aus heutiger Sicht)
    - schwach (aus heutiger Sicht)
    - Analyse Ciphertext allein ist durchaus erfolgversprechend (Prinzip)

4.3 Mittel der Kryptographie (3)

- Verschlüsselung
  - S-Box
    - .....
    - N → S-Box → M
    - Realisierung in modernen Verfahren
    - linear, M=N
    - nichtlinear, N<M
    - nichtlinear, N>M → schwache Einwegefunktion
    - mehrfache Kaskadierung solcher S-Boxen

4.3 Mittel der Kryptographie (4)

- Verschlüsselung
  - Permutation (Transposition)
    - Wechsel der Zeichenpositionen
    - Beispiel Spaltentransposition



- Am Wirkungsvollsten bei Permutation auf Bitebene (warum?)

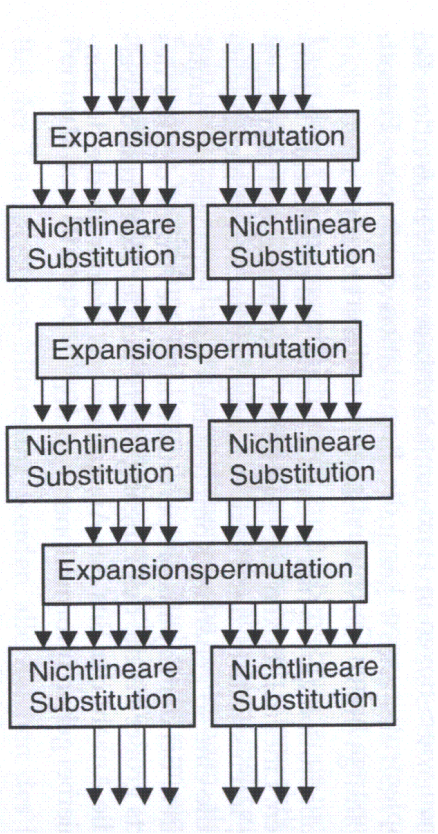
4.3 Mittel der Kryptographie (5)

- Verschlüsselung
  - P-Box
    - .....
    - N → P-Box → M
    - Realisierung in modernen Verfahren
    - N Positionen Eingangswert kommen auf M Positionen Ausgangswert
    - linear, M=N
    - nichtlinear, N<M → Expansionspermutation
    - nichtlinear, N>m → Kompressionspermutation → Einwegefunktion

Quelle: Manfred Lipp: VPN – Aufbau und Sicherheit

4.3 Mittel der Kryptographie (6)

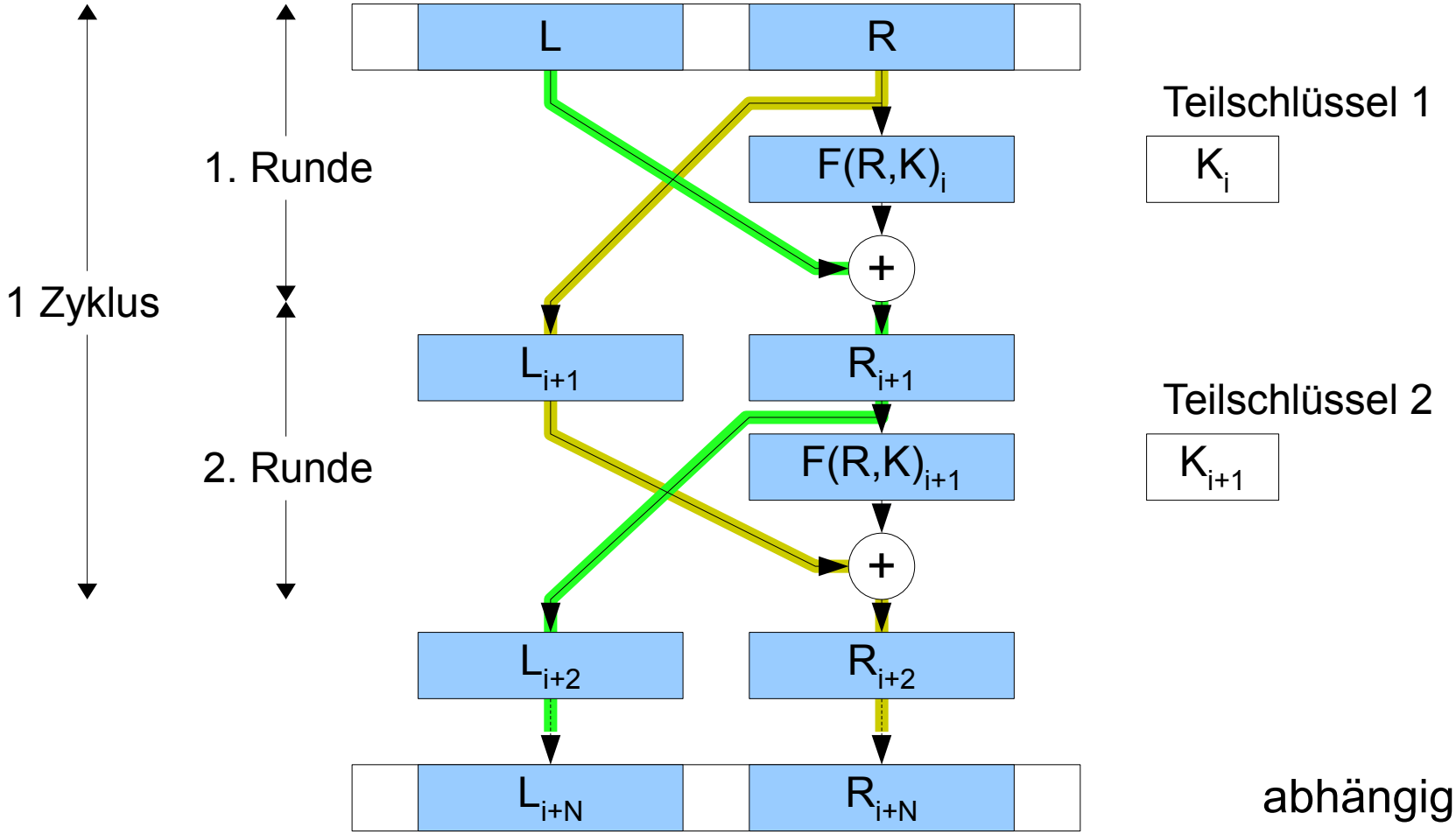
- Verschlüsselung
  - mehrfache Kaskadierung von S-Boxen und P-Boxen



Quelle: Manfred Lipp: VPN – Aufbau und Sicherheit

4.3 Mittel der Kryptographie (7)

- Verschlüsselung
  - Feistel-Netzwerk, Feistelzyklus



4.3 Mittel der Kryptographie (8)

- Verschlüsselung
  - Feistel-Netzwerk, Feistelzyklus
    - DES nutzt 8 Zyklen = 16 Runden
    - offengelegt → gründlich getestet
    - Entschlüsselung durch den gleichen Zyklus mit vertauschten Teilschlüsseln und L und R (Ablauf)
    - symmetrische Verschlüsslung

4.3 Mittel der Kryptographie (9)

- XOR (....., ...)
  - XOR mit Zufallsfolge beseitigt / verdeckt statistische Abhängigkeiten
  - XOR leicht ausführbar
  - XOR leicht umkehrbar (Beispiele)
- Modulo-Arithmetik
  - $a = b\,(mod\ n)$ falls $a = b + k \cdot n$ mit $n - natürliche\ Zahl$ und $a \geqslant 0$ und $0 \geqslant b \geqslant n$
  - kommutativ, distributiv, assoziativ
  - $(a+b)\ mod\ n = ((a\ mod\ n) + (b\ mod\ n))\ mod\ n$
  - $(a \cdot b)\ mod\ n = ((a\ mod\ n) \cdot (b\ mod\ n))\ mod\ n$ (Beispiele)
  - $A = g^x\ mod\ n$ falls $x; n$ hinreichend groß $\Rightarrow$ Einwegefunktion
    auch bei bekannten A, g, n ist x nicht berechenbar
  - praktisch z. B. n: Primzahl mit mindestens 768 bit Länge (Primzahl, da...)
    x: mindestens 160 Dezimalstellen

4.3 Mittel der Kryptographie (10)

- Primzahlen
  - nicht zerlegbar
  - $a = b \cdot c$ $\quad b\ ;\ c$ - Primzahlen
    Mit hinreichend großen b und c ist das eine Einwegefunktion.

- Zufallszahlen
  - schwierig zu erzeugen im Rechner
  - Qualität entscheidend für Sicherheit Verfahren (z. B. x auf Seite zuvor)

- Schlüssel
  - immer zumindest teilweise geheim
  - nicht zurückrechenbar
  - Länge / Längenreserve → Diskussion

- hier nicht abschließend, nur Auswahl wichtiger Mittel

4.3 Mittel der Kryptographie (11)

- Schlüssel
  - immer zumindest teilweise geheim
  - nicht zurückrechenbar
  - Länge / Längenreserve → Diskussion

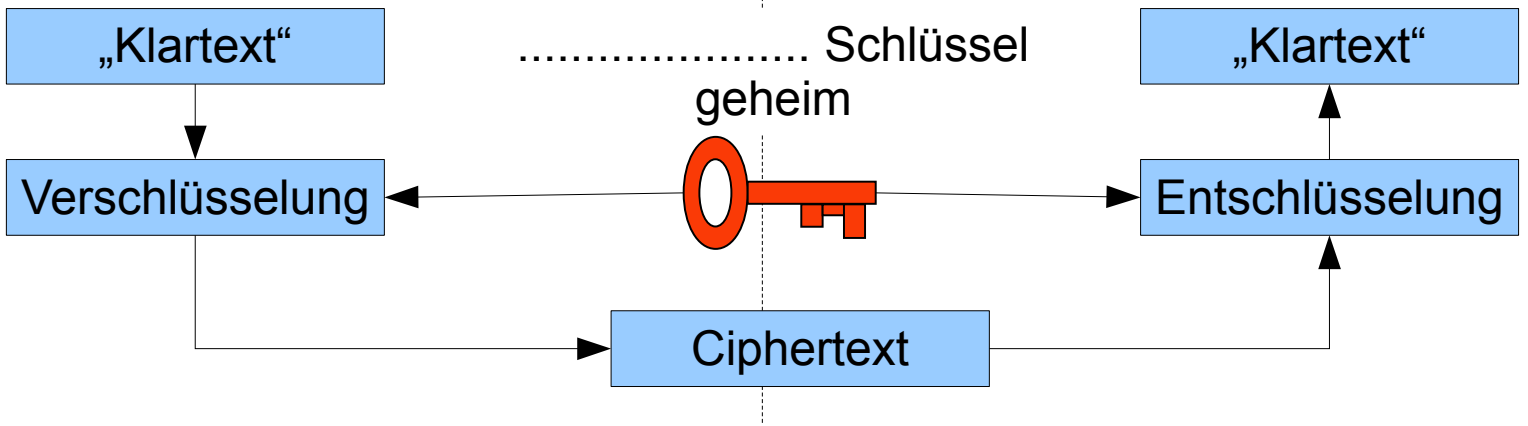
Schlüsselart	Länge	Mögliche Kombinationen	Test-zeit	Parallele Tests	Mittlere Suchzeit
Kofferschloss	10 Bit	1.000	2 s	1	17 Minuten
Kurzes Pass-wort	28 Bit	8.1450.625	1 μs	1	40 Sekunden
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 μs	1	6 Tage
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 μs	50	2,8 Stunden
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 μs	1.000.000	0,05 Sekun-den
DES-Schlüssel	56 Bit	72.057.594.037.927.900	1 μs	1	1.142 Jahre
DES-Schlüssel	56 Bit	72.057.594.037.927.900	1 μs	1.000.000	10 Stunden
IDEA-Schlüssel	128 Bit	$3{,}4 * 10^{38}$	1 μs	1	$5{,}4 * 10^{24}$ Jahre
IDEA-Schlüssel	128 Bit	$3{,}4 * 10^{38}$	1 μs	1.000.000	$5{,}4 * 10^{18}$ Jahre

Quelle: Manfred Lipp: VPN

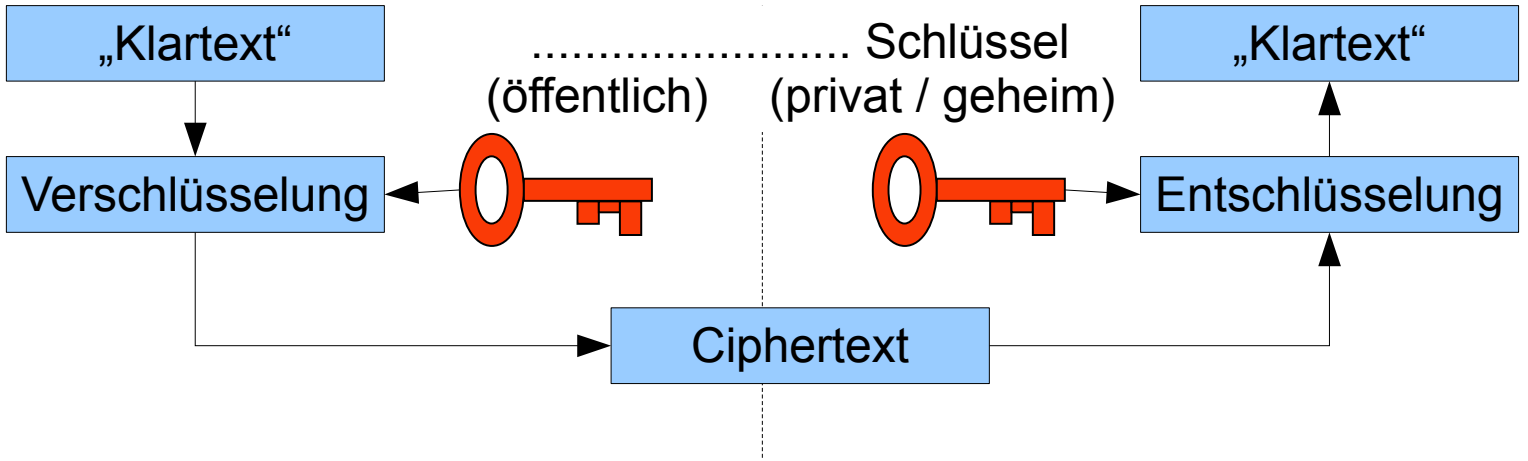
- hier nicht abschließend, nur Auswahl wichtiger Mittel

4.4 Einige Verfahren (1)

- **Verschlüsselung**
  - ..... Verschlüsselung

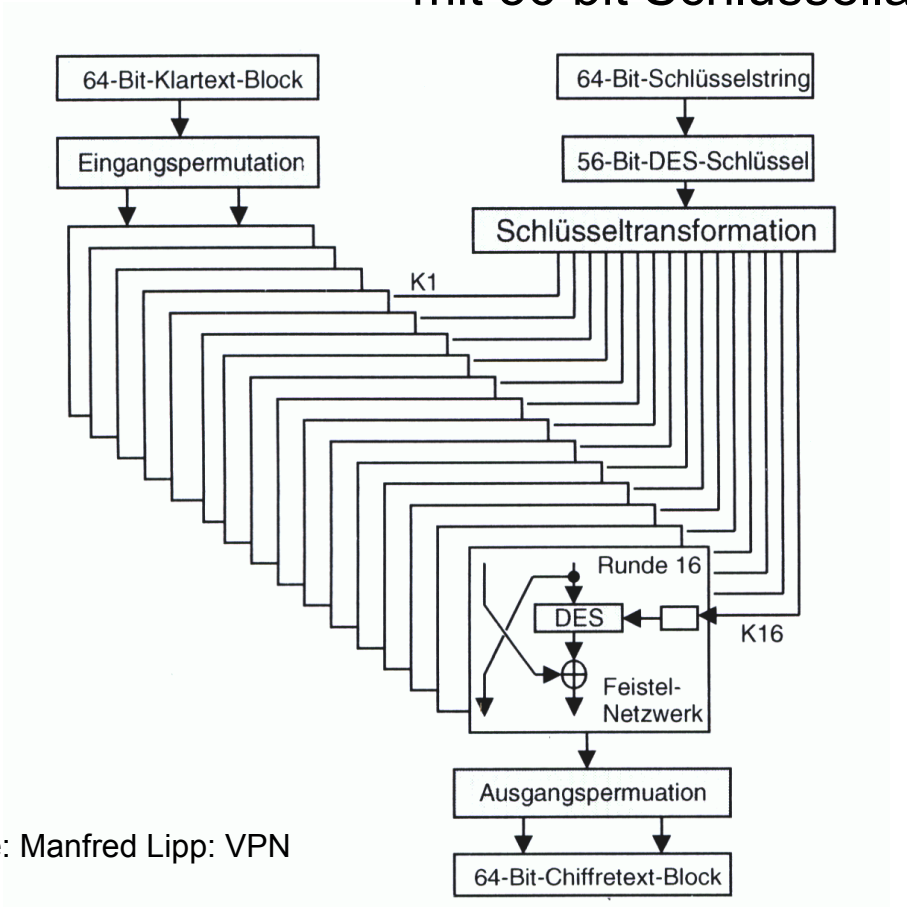


  - ..... Verschlüsselung

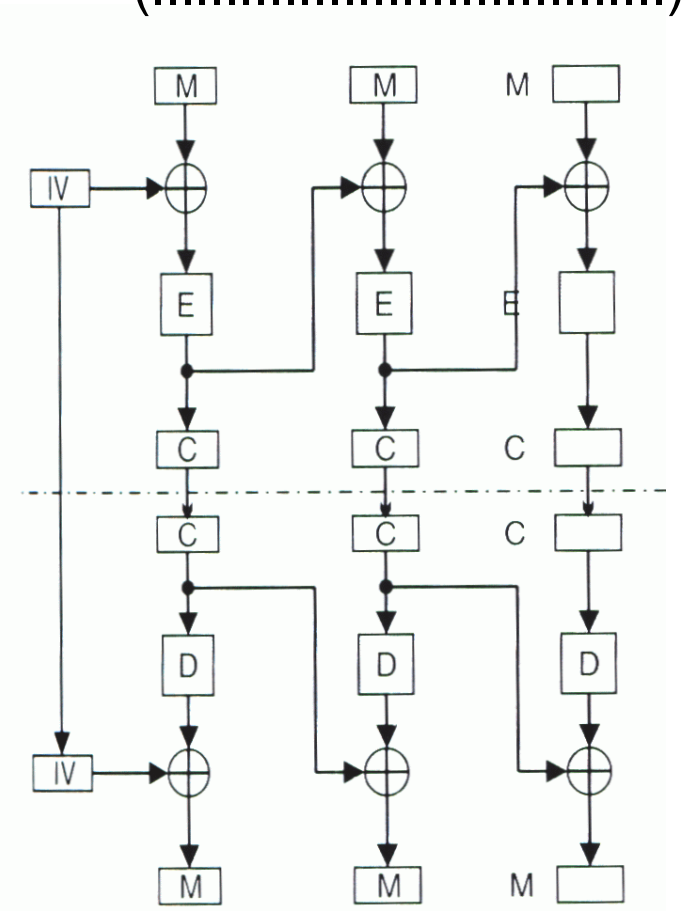


4.4 Einige Verfahren (2)

- Verschlüsselung
  - symmetrische Verschlüsselung
    Beispiel: DES – beruht auf Feistel-Algorithmus – erweitert mit CBC
    mit 56 bit Schlüssellänge (.....)



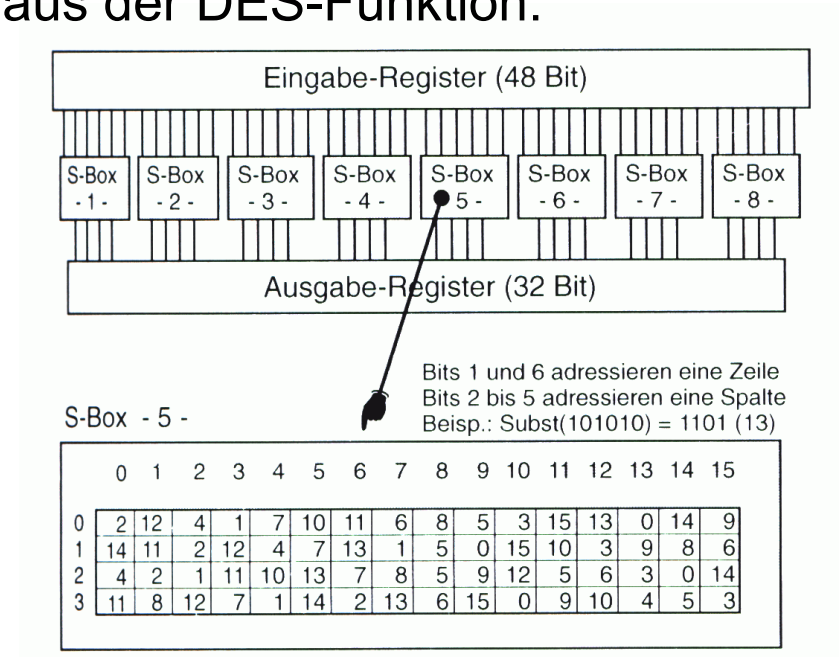
Quelle: Manfred Lipp: VPN



4.4 Einige Verfahren (4)

- Verschlüsselung
  - symmetrische Verschlüsselung - Beispiel: DES
    - CBC als Mittel gegen differentielle Cryptoanalyse
      1974 seitens IBM oder NSA hinzugefügt

      → Gleiche Klartextblöcke gehen nicht gleich in die Kodierung ein.
    - Beispiel aus der DES-Funktion:

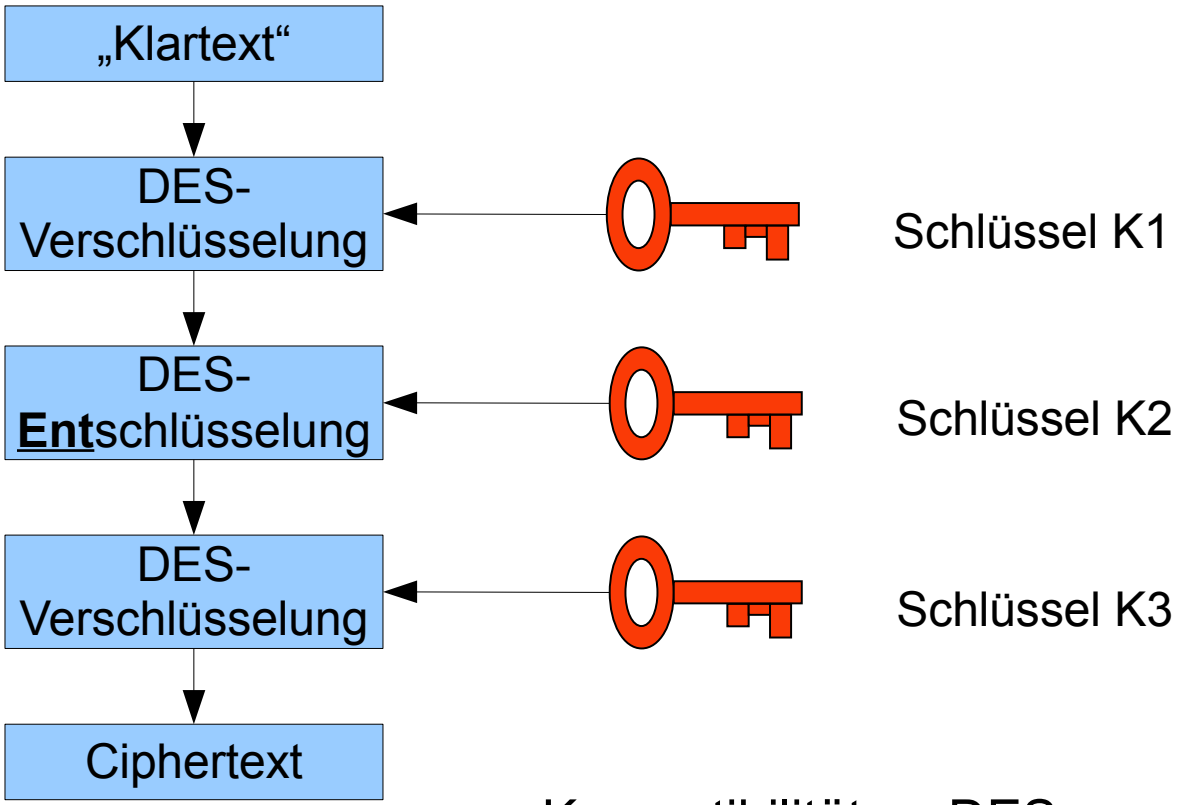


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	5	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Quelle: Manfred Lipp: VPN

4.4 Einige Verfahren (5)

- Verschlüsselung
  - symmetrische Verschlüsselung - Beispiel: Triple-DES
    - Schlüssellänge 3 x 56 Bit = 168 Bit



Kompatibilität zu DES:.... = ... = ...

4.4 Einige Verfahren (6)

- Verteilung / Aushandlung symmetrischer Schlüssel
  - manuelle Verteilung
  - Verschicken auf sicheren (?) Wegen (Henne-Ei-Problem)
  - Schlüsselaustausch / Erzeugung symmetrischer Schlüssel mithilfe der Public Key Kryptographie (siehe dort)
  - Erneuerungsrate

Diskussion

4.4 Einige Verfahren (7)

- Verschlüsselung
  - Schlüsselaustausch mit dem Diffie-Hellman-Verfahren (symm. Schlüssel)
    - Nutzung der Einwegefunktion $z = g^x \ mod \ n$

Teilnehmer A	Teilnehmer B
1.) erzeugt x (große Zufallszahl)	1.) erzeugt y (große Zufallszahl)
2.) $a = g^x \ mod \ n$	2.) $b = g^y \ mod \ n$
3.) schickt a (; g; n) an B	3.) schickt b (; g; n) an A
4.) $k = b^x = (g^y \ mod \ n)^x = g^{y \cdot x} \ mod \ n$	4.) $k' = a^y = (g^x \ mod \ n)^y = g^{x \cdot y} \ mod \ n$
	$k = k'$
5.) x ..... vernichten	5.) y ..... vernichten

Vertraulichkeit von x und y!!!

4.4 Einige Verfahren (8)

- Verschlüsselung
  - unsymmetrische Verschlüsselung – Prinzip
    - öffentlicher Schlüssel → nur zum Verschlüsseln
      Einwegefunktion – auch zum Entschlüsseln geeignet?
    - privater Schlüssel: enthält Zugang zu Falltür in der Einwegefunktion;
      → nur zum Entschlüsselung – auch zum Verschlüsseln geeignet?
    - erster Ansatz durch Whitfield Diffie und Martin Hellman (1976):
      erst einmal nur Verfahren zum Austausch symmetrischer Schlüssel
      (Henne-Ei-Problem)
    - weiterer Ansatz durch Dr. Ronald Rivest, Adi Shamir und Leonard Adleman (1977) → RSA-Verfahren
      Damit waren Austausch von symmetrischen Schlüsseln und die Verschlüsselung selbst möglich.

4.4 Einige Verfahren (9)

- Verschlüsselung
  - (asymmetrisches) RSA-Verfahren
    - Basis sind ein Produkt zweier sehr großer ..... und die praktisch nicht mögliche .....
    - Modulorechnung anwenden
    - $M = (M^e)^d \ mod \ n$ passende Paare von ... und ... finden!!!!!
      Public Key: {n,e}
      Private Key: {n,d}
    - komplizierte Zusammenhänge, die zur Lösung entsprechend obiger Gleichung führen
    - Verschlüsseln mit dem Public Key {n,e}
    - Entschlüsseln mit dem Private Key {n,d}
    - (Verwendung zum Signieren – Vorgriff, Ablauf)

4.4 Einige Verfahren (10)

- **HASH-Funktionen**
  - Einwege-.....
  - „Fingerprint“ einer Datenzusammenstellung
  - schnell und einfach aus beliebig langem Eingangswert M Hashwert h mit Größe n bildbar
  - aus n ist M nicht berechenbar
  - zu M praktisch keine M' mit identischem h erzeugbar; ist das Ziel
  - kleine Änderungen in M → große Änderungen in h

(Diskussion der Möglichkeiten)

4.4 Einige Verfahren (11)

- HASH-Funktionen
  - Beispiel: MD5 (128 bit), SHA (160; 224; 265; 384; 512 bit), RIPEMD (128; 160; 256; 320 bit)
    - 2004 ist für MD5 Verfahren bekannt geworden, um Kollissionen zu erzeugen
    - 2005/6 für SHA-1 (160 bit) ist Verfahren zur Berechnung von Kollisionen **mit 25% sinnvollem Inhalt** bekannt geworden!!!

      (Diskussion der Konsquenzen)

  - Verwenden von h (Sichern von h durch Schlüssel) → Beweis der Integrität (?)

4.4 Einige Verfahren (12)

- **Authentifizierung**
  - starke Authentifizierung → Nachweis der Identität bei .....
    .....
  - schwache Authentifizierung → eine Person hat und / oder weiß etwas, was normalerweise nur eine bestimmte Person hat und / oder weiß.
  - Verbesserung der schwachen Authentifizierung wird angestrebt
    - „something to have and something to know“
    - biometrische Daten
    - einmalige persönliche Prüfung im Zusammenhang mit den anderen

4.4 Einige Verfahren (13)

- **Verwaltung von Public Keys**
  - Wie bekomme ich den öffentlichen Schlüssel einer anderen Person?
  - Wie bekomme ich diesen Schlüssel zuverlässig, wirklich den dieser Person?

  - persönlicher Austausch - direct trust
  - Austausch auf sicherem (?) Weg
  - Auf unsicherem Weg und Vergleich Fingerprint (.....)
  - Weitergabe von Nutzer zu Nutzer mit Vererbung der Vertrauensstellung (dezentrales Verfahren)
  - Von Zentraler Instanz, zu der ich einen sicheren Kanal habe (zentrales Verfahren), der ich vertraue

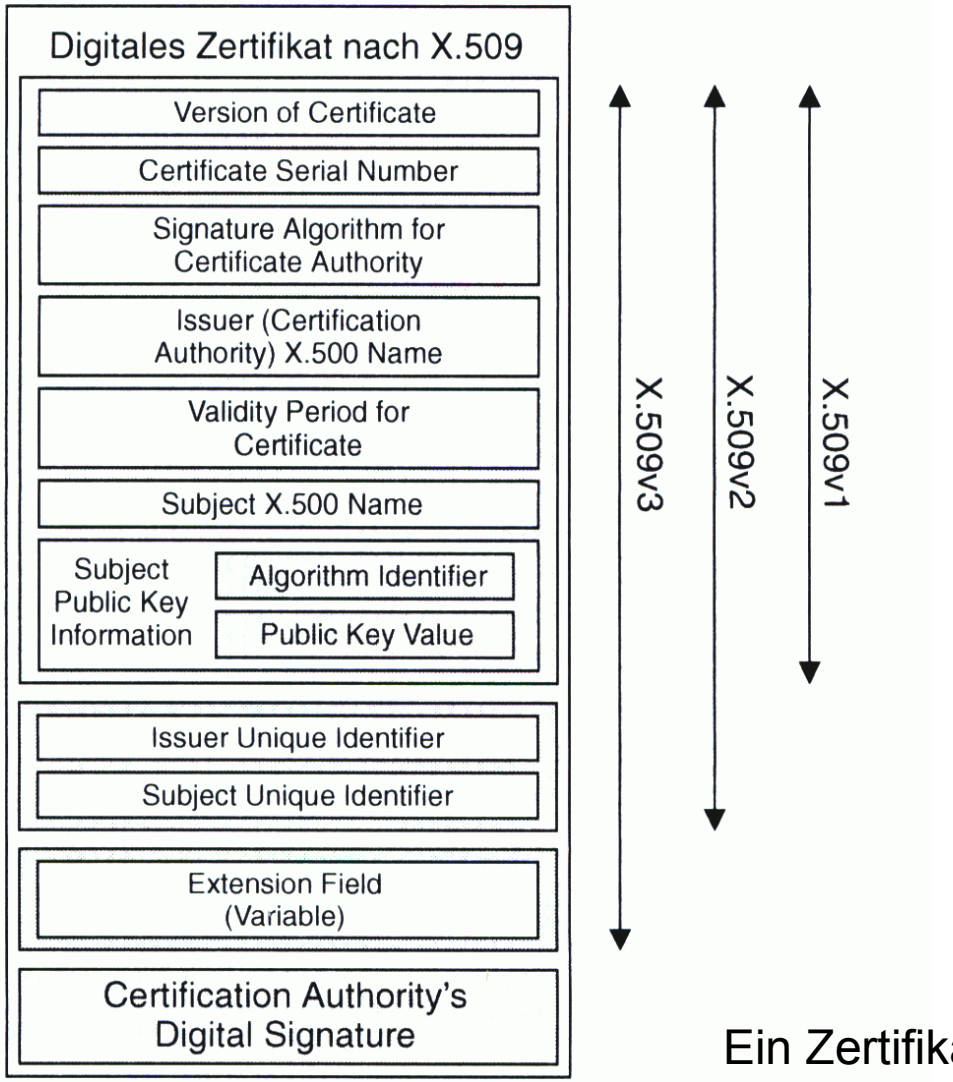
  - Es ist immer eine Frage der Vertrauensstellung zum Geber des Schlüssels und der Sicherheit des Übermittlungskanals

4.4 Einige Verfahren (14)

- Verwaltung von Public Keys
  - Zertifikate
    - Verbindung des öffentlichen Schlüssels mit Daten zum Schlüsselbesitzer, von einer mehr oder weniger zentralen, mehr oder weniger vertrauenswürdigen Stelle geprüft und signiert
      - Angaben zum Besitzer
      - organisatorische Daten
      - Angaben zur ausstellenden Stelle
      - öffentlicher Schlüssel
      - Signatur, durch ausstellende Stelle
    - Signatur der (mehr oder weniger) zentralen Stelle ist vergleichsweise leicht prüfbar

4.4 Einige Verfahren (15)

- Verwaltung von Public Keys
  - Zertifikate



Ein Zertifikat nach ITU-X.509
(Vergleich mit realem Zertifikat)

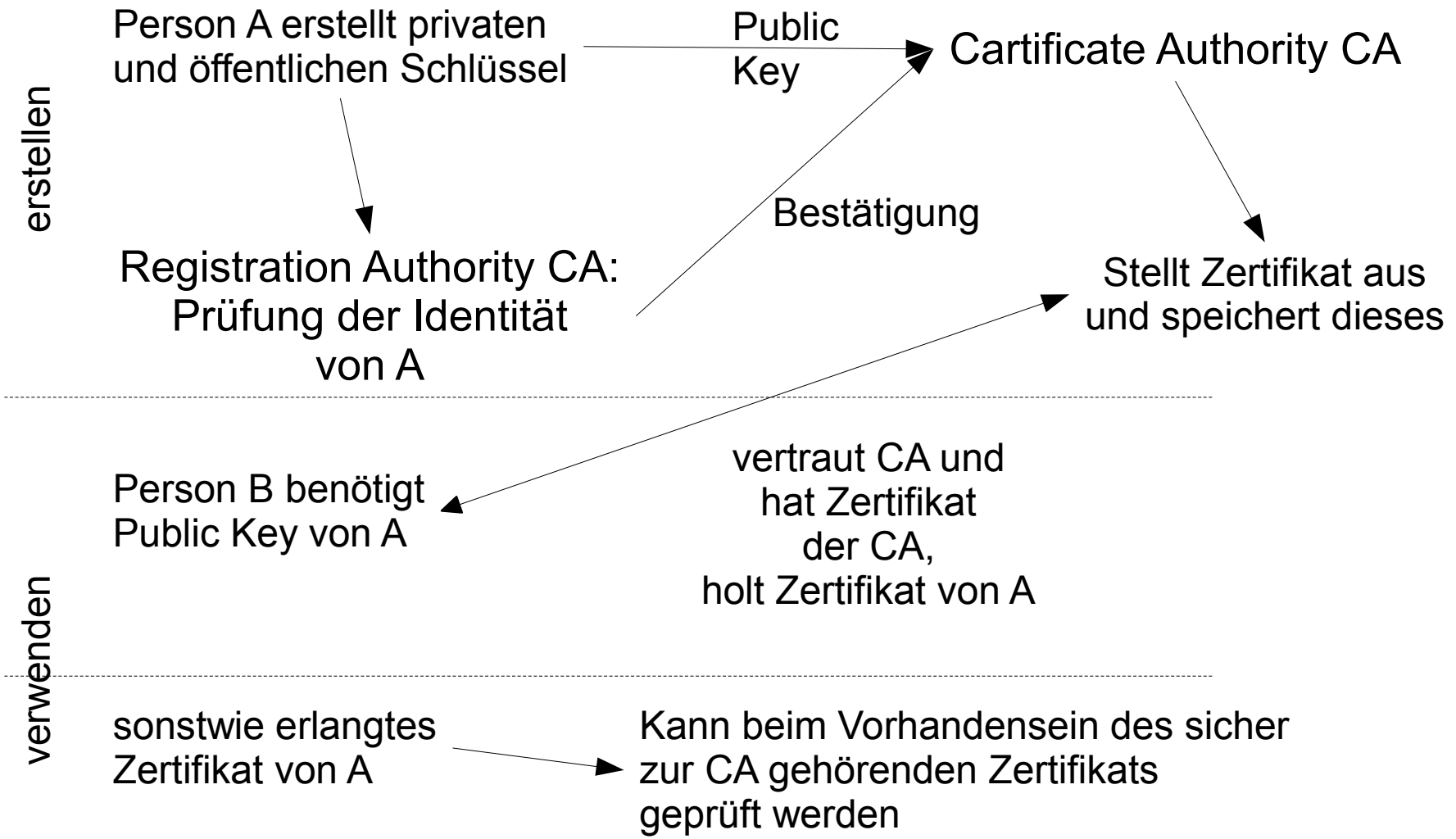
Quelle: Manfred Lipp: VPN

4.4 Einige Verfahren (16)

- **PKI – Public Key Infrastructure / öffentlich oder privat**
  - Vertrauensmodell
  - Certificate Authority (CA)
  - Registration Authority (RA)
  - Zertifikat-Management
    - Verteilen
    - Zugang, Protokolle für Anforderung und Verteilung
    - Überwachung der Lebensdauer
    - Sperrungen

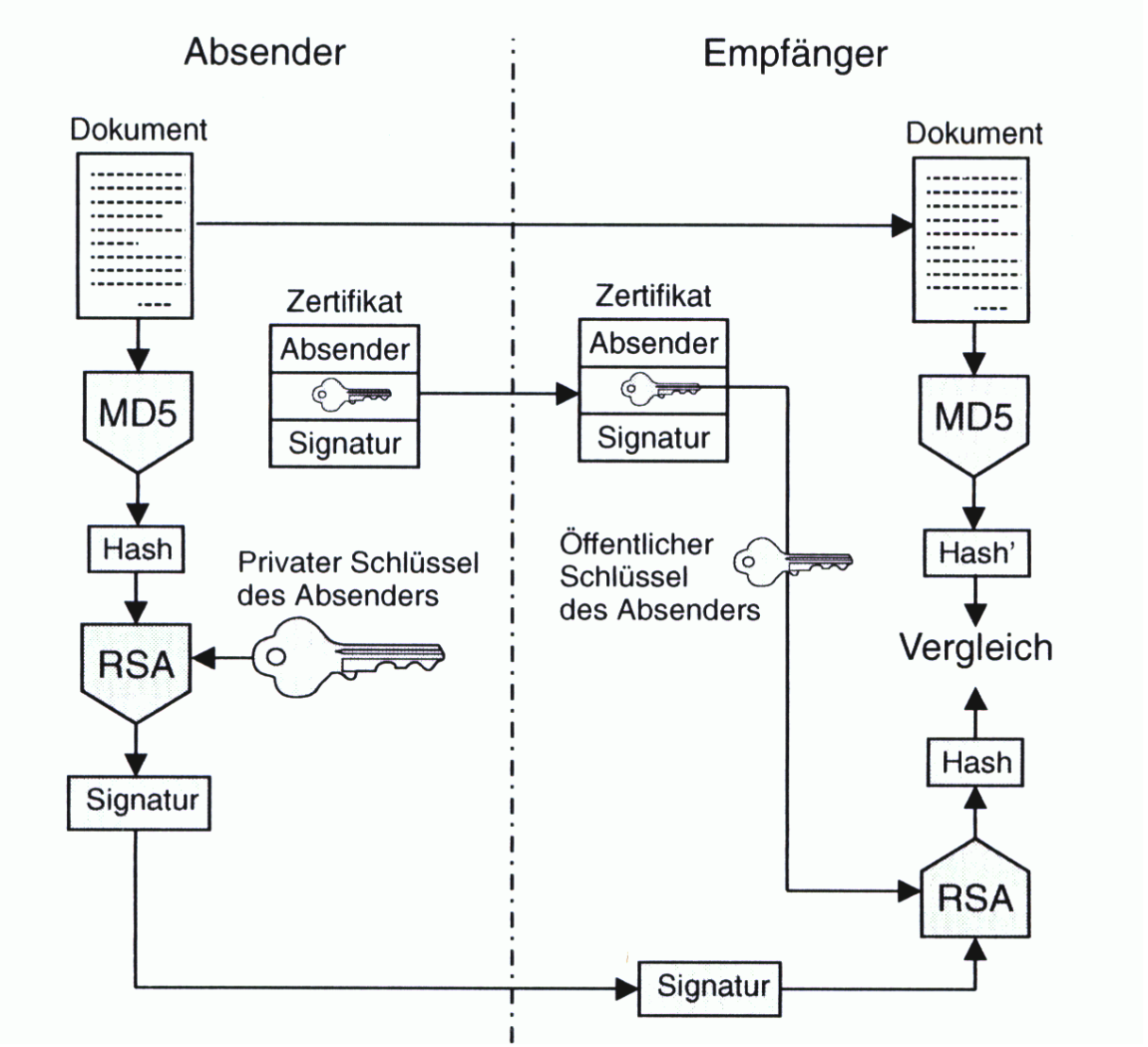
4.4 Einige Verfahren (17)

- PKI – Public Key Infrastructure – Erstellen / Ausgeben und Verwenden des Zertifikates



4.4 Einige Verfahren (18)

- PKI – Public Key Infrastructure – Beispiel für Anwendung Signatur



Quelle: Manfred Lipp: VPN

Eine digitale Signatur mit MD5 und RSA

4.4 Einige Verfahren (19)

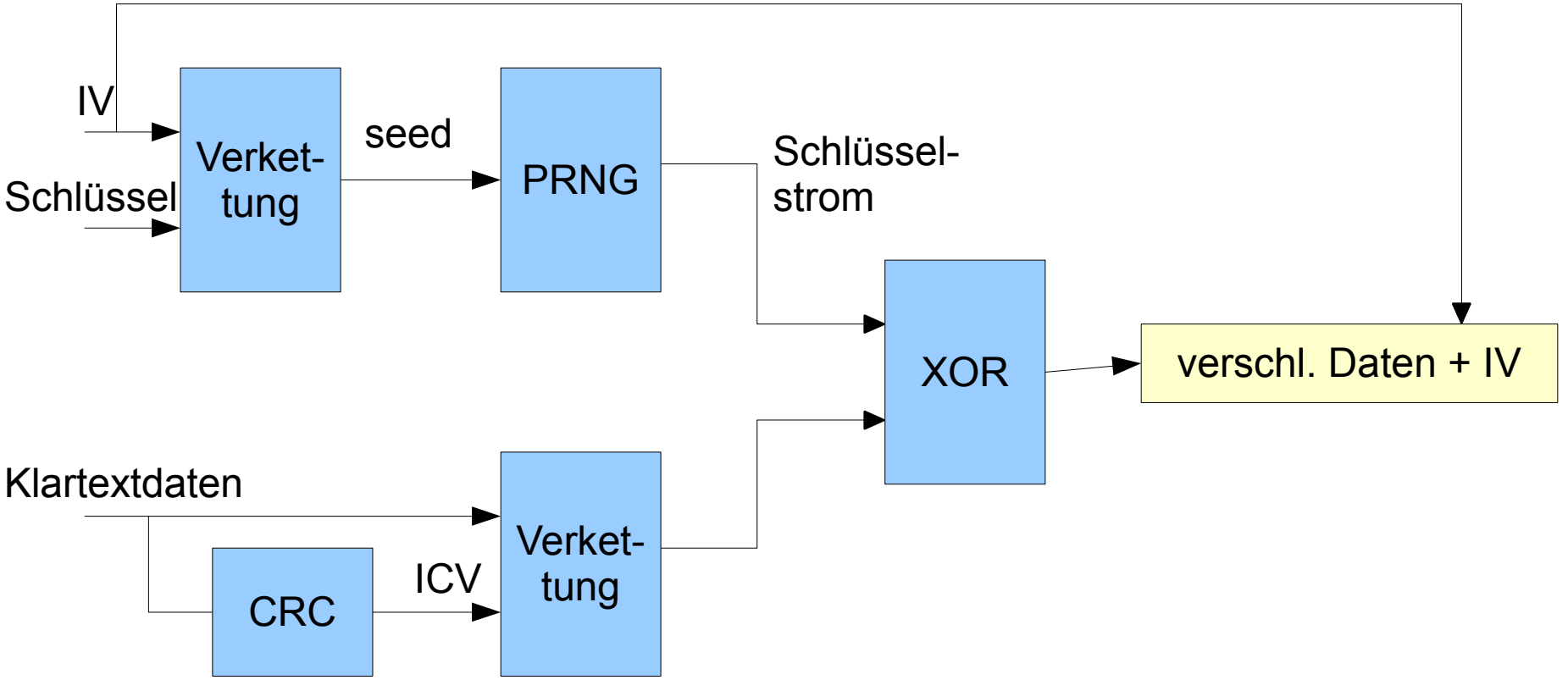
- Und hier ein Beispiel, wie man es nicht machen sollte: WEP

  WEP (WLAN, ...):

  - symmetrische Stromverschlüsselung auf Schicht 3
  - Schlüssellängen 40 (64) und 104 (128) Bit
  - RC4 Pseudozufalls(zahlen)generator - PRNG (von RSA)
  - Schlüssel + Initialisierungsvektor → PRNG → Schlüsselstrom
  - Klartextdatenstrom mit CRC gegen Fehler gesichert
  - Klartextdatenstrom XOR Schlüsselstrom = verschlüsselter Datenstrom
  - IV zusammen mit verschlüsselten Daten an den Empfänger

4.4 Einige Verfahren (20)

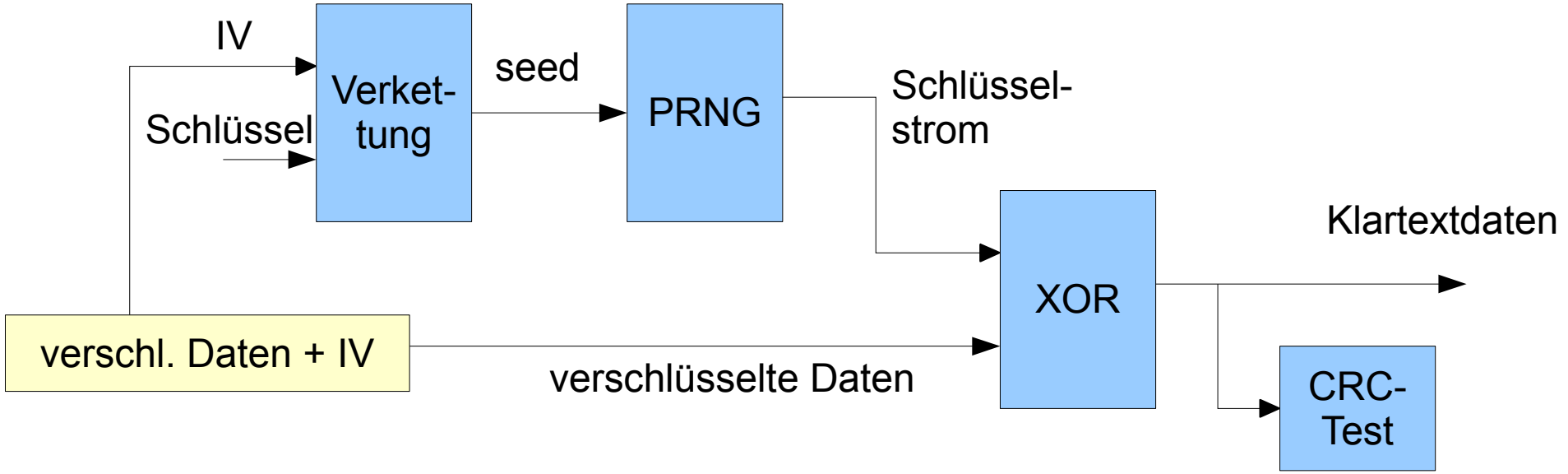
- WEP – Verschlüsselung (Encodierung)



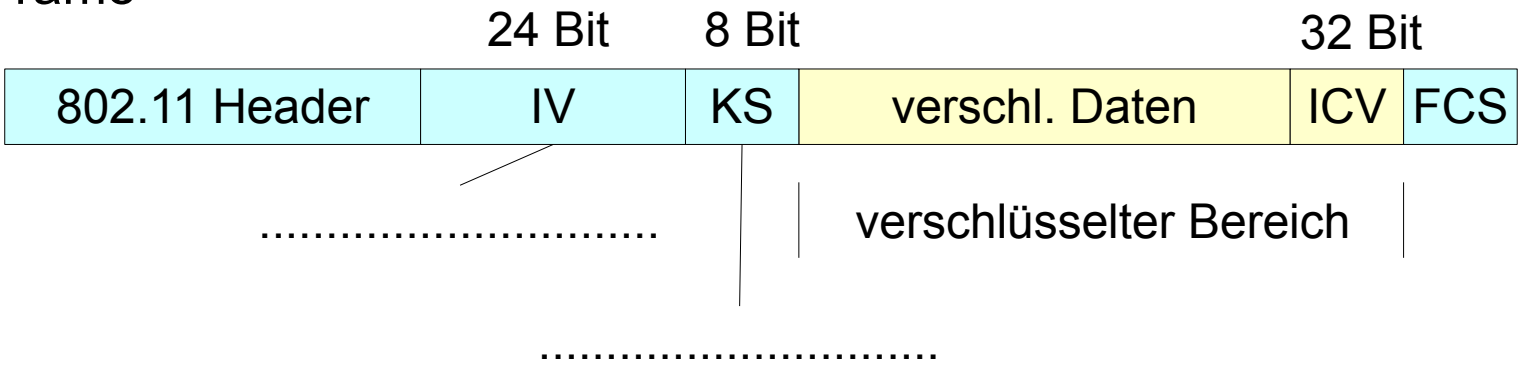
ICV – .....
IV – .....

4.4 Einige Verfahren (21)

- WEP – Entschlüsselung (Decodierung)



- Frame



4.4 Einige Verfahren (22)

- WEP – Schwachstellen
  - Schlüsselmanagement (Verteilung, Erneuerung)
  - Schlüssellänge (40 Bit)
  - mangelhafter Schutz vorm Mitlesen (Rückschlüsse auf Klartext)
  - Authentifizierung (Wer ist der Partner? Ist der Anmelder berechtigt?)
  - Zugriffskontrolle (Wer darf Nachrichten Einspeisen?)
  - Sicherheit geheimer Schlüssel (Keine analytische/ statistische Ermittlung)

  jetzt zu den einzelnen Arten der Schwachstellen eine Auswahl an Beispielen, nicht vollständig.

4.4 Einige Verfahren (23)

- Schlüsselmanagement (Verteilung, Erneuerung)

?

4.4 Einige Verfahren (24)

- Schlüssellänge

  - 40 bit (.....)
    40 Bit SK + 24 Bit IV → 64 Bit K

    40 Bit = 1.099.511.627.776
    nach Tabelle von ca. 2000 bei Brute Force Stunden bis Tage
    Schlüssel wird in der Regel über Wochen oder Jahre nicht geändert
    (siehe Punkt zuvor)

  - Später 104 Bit
    104 Bit SK + 24 Bit IV → 128 Bit K
    schon besser, beseitigt aber nicht die anderen Probleme

4.4 Einige Verfahren (25)

- mangelhafter Schutz vorm Mitlesen (Rückschlüsse auf Klartext) (1)

  - 2^{24} IV möglich: 16.777.216
    bei im Mittel 1000 Byte Nutzdaten/ Paket und 5 Mbit/s → ca. 7,5 h

  - Bei rein zufälliger Auswahl des IV (z. B. viele Stationen) im Durchschnitt innerhalb 4.096 Paketen zwei gleiche IV

  - Wiederholung von Schlüsselströmen (selber IV und selber geheimer Schlüssel SK) → Rückschlüsse auf Klartexte
    hier binär, gilt auch allgemein (Beispiel: Text + Text / Text - Text))

  - Sind durch „known plain text“ Teile von Schlüsselströmen bekannt, kann bei der Wiederverwendung des Schlüssels K weiterer Klartext gewonnen werden (Beispiel LLC/SNAP-Header: 8 Byte).

4.4 Einige Verfahren (26)

- mangelhafter Schutz vorm Mitlesen (Rückschlüsse auf Klartext) (2)

  - Da es ein Stromverschlüsselungsverfahren ist, kann aus der Länge der ..... Nachricht EM auf die Art der ..... Nachricht P geschlossen werden. Das allein muss noch kein Problem sein, kann es aber im Zusammenhang mit anderen Schwächen werden.

4.4 Einige Verfahren (27)

- Authentifizierung (Wer ist der Partner? Ist der Anmelder berechtigt?)

  - Alle nutzen den selben geheimen Schlüssel und authentifizieren sich darüber. → Jeder kann sich als anderer ausgeben.

  - Beim Mitlesen einer Authentifizierung können Daten gewonnen werden, die ein unberechtigtes Authentifizieren ermöglichen. Zitat: „Das shared Key Authentication ist die sichere Variante...“ (das ELektronik KOmpendium – im Internet) – Ist das ernst gemeint?

  - Beim Mitlesen einer Authentifizierung kann ein (kurzer) Schlüsselstrom KS gewonnen werden, der bei der erneuten Verwendung des IV mit dem selben SK zur Dekodierbarkeit von Teilen des Klartextes führt.

    (Beispiel)

4.4 Einige Verfahren (28)

- Zugriffskontrolle (Wer darf Nachrichten Einspeisen?)

  - Siehe Authentifizierung bzw. das, was man dafür hielt

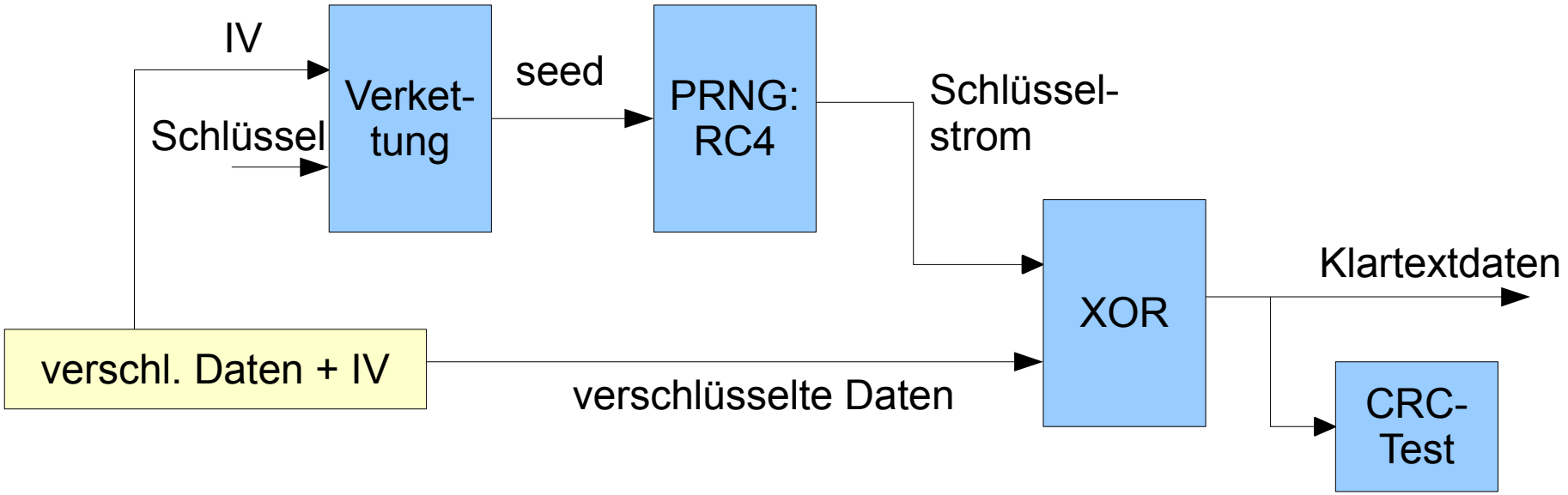
  - Mit der Kenntnis von einzelnen Schlüsselströmen kann (gezielter) Unfug in das Netz eingespeist werden. Es kann z. B. ein bei einer mitgehörten Authentifizierung gewonnener Schlüsselstrom verwendet werden.

  - Mitgelesene verschlüsselte Nachrichten, auf deren Funktion geschlossen werden kann (siehe „mangelhafter Schutz vorm Mitlesen – 2“) können gezielt in das Netz eingespeist werden, ohne dass formal ein Fehler erkannt werden kann. Mit diesen eingespeisten Daten kann das Netz zur Preisgabe von Informationen gebracht werden (z. B. ARP-Pakete → AP sendet diese mit einem anderen IV erneut aus.)

4.4 Einige Verfahren (29)

- Sicherheit geheimer Schlüssel (keine analytische/ statistische Ermittlung)

  - Schwäche von RC4 - speziell in dieser Anwendung



  - Wenn es gelingt, aus dem bekannten Teil des gesamten Schlüssels (IV) eine statistische Abhängigkeit zwischen Bytes des Schlüsselstroms und dem geheimen Schlüssel herzustellen und
  - Wenn an der entsprechenden Stelle des Klartextes ein bekannter Wert steht ja, dann... (Skizze)

4.4 Einige Verfahren (30)

- Schwachstellen RC4 – in dieser Verwendung

  KSA (Key Scheduling Algorithm)
  + PRGA (Pseudo Random Generator Algorithm) = PRNG

  - KSA: Tabelle mit $N=2^n$ Werten (Zeilen), je Wert (Zeile) n Bit
    - Tabelle mit Werten füllen: 0,1,2,3,4,..., N-1
    - i = 0 j = 0 S[i] = i
    - Schleife für i=0 bis N-1
      j =[j + S[i] + K[i mod l]] mod N
      swap (S[i], S[j])

  - PRGA: Auf Basis der Matrix aus KSA Schlüsselstrom erzeugen
    - i = 0 j = 0
    - Schleife
      i = i + 1 j =[j + S[i]] mod N
      swap (S[i], S[j])
      Ausgabe z = S[S[i] + S[j]] → Keystream

4.4 Einige Verfahren (31)

- Schwachstellen RC4 – in dieser Verwendung (Fortsetzung)

  RC4 muß eine ..... realisieren. Das klappt eigentlich auch, zumindest in Näherung, solange kein Anteil des Schlüssels bekannt ist.

  Leider sind jedes mal die ersten drei Byte eines Schlüssels bekannt, der IV.

  Bestimmte IV, die ja erkennbar sind, bringen mit erhöhter Wahrscheinlichkeit (ca. 5...15 %) Werte, die mit bestimmten Bytes des geheimen Schlüssels korrelieren, an vordere Stellen des Schlüsselstroms. Dort ist mit hoher Wahrscheinlichkeit der Klartext bekannt (LLC/SNAP).

  Über „verschlüsselter Text“ ..... „bekannter Klartext“ ist der Schlüsselstrom an der Stelle mit hoher Wahrscheinlichkeit ermittelbar.

  Wenn der Schlüsselstrom an solchen Stellen Rückschlüsse auf Teile des geheimen Schlüssels zulässt, dann ist der geheime Schlüssel nicht mehr geheim.

  Man braucht immer eine bestimmte Anzahl Pakete, die mit bestimmten IVs verschlüsselt wurden.

4.4 Einige Verfahren (32)

- Schwachstellen RC4 – in dieser Verwendung (Fortsetzung)

  Insbesonder das Verfahren nach Fluhrer, Mantin und Shamir:

  Wenn zu einer bestimmten Menge von IV die jeweils ersten Bytes des Schlüsseldatenstroms bekannt sind, dann können mit relativ wenig Aufwand und mit recht hoher Wahrscheinlichkeit bestimmte Bytes des geheimen Schlüssels ermittelt werden.

4.4 Einige Verfahren (33)

Übertragungstechnik I und II

5 Kanalkodierung
- 5.1 zur Einordnung
- 5.2 Ziele
- 5.3 Begriffe, Festlegungen und mathematische Mittel
- 5.4 einige Kodes und Verfahren

5 Kanalkodierung
5.1 zur Einordnung (1)

- Überschneidungen mit der Kanalkodierung
  → deshalb beides hier kurz vorgestellt:

<u>Sicherheitskodierung</u>

- Schutz vor
  - ..... Beeinflussung der Nachricht
  - (..... Beeinflussung der Nachricht)
  - unberechtigter Kenntnisnahme des Inhalts der Nachricht

<u>Kanalkodierung</u>

- Schutz vor
  - (..... Beeinflussung der Nachricht)
  - ..... Beeinflussung der Nachricht

- Informationstheorie -
- Mathematik -
- Kryptographie -

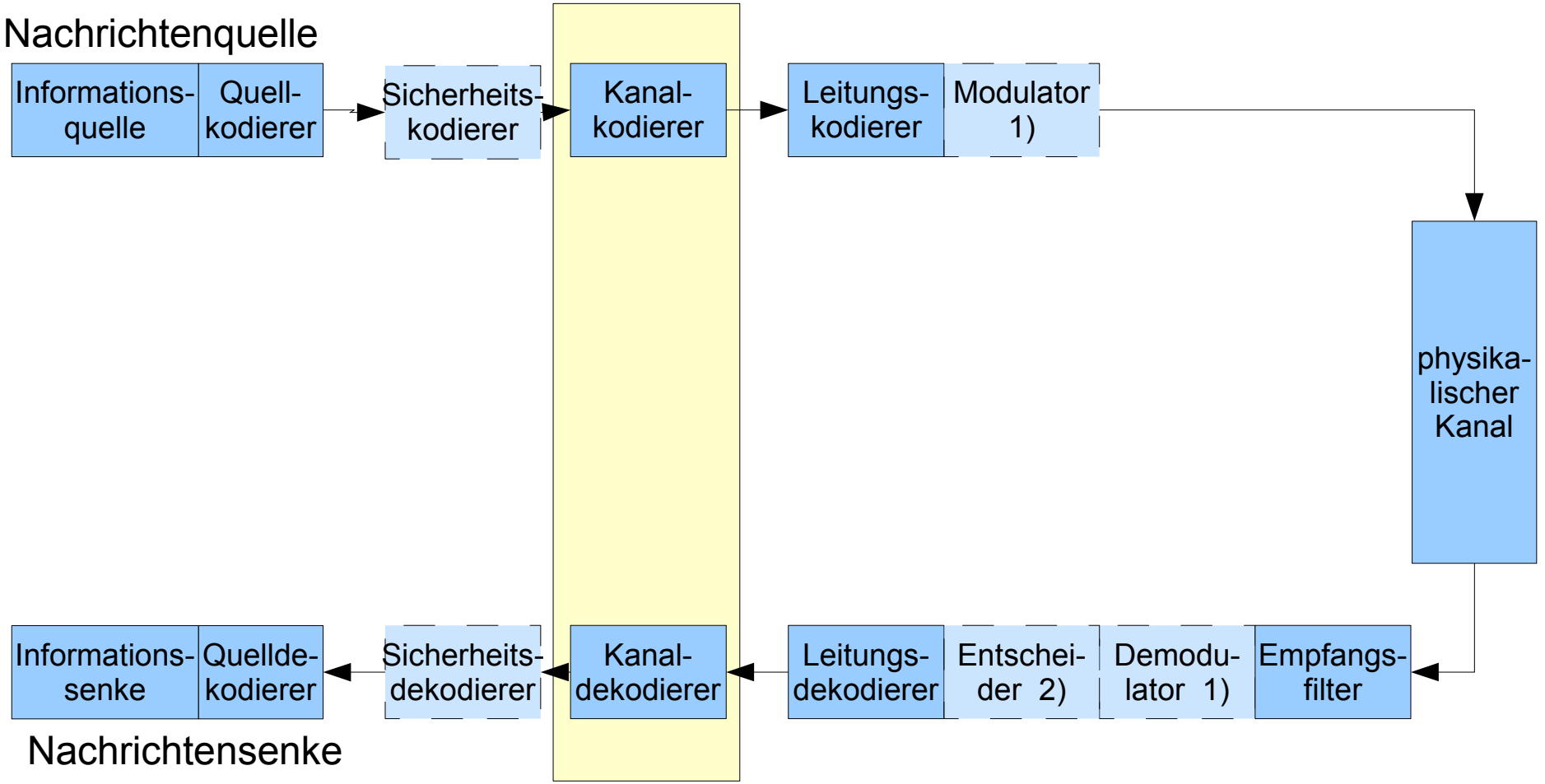
kommt mehr aus der Datenwelt

kommt mehr von der Untersuchung von Kanälen

5.1 zur Einordnung (2)

- Die Kanalkodierung befindet sich im Signalfluß, aus Richtung der Quelle gesehen, vor der .....
- In einem Teil der Fälle leistet die Sicherheitskodierung schon alles das mit, was für die Übertragung des Signals sonst in der Kanalkodierung realisiert wird.
- In einem weiteren Teil der Fälle werden Sicherheitskodierung und Kanalkodierung nacheinander angewendet.
- Da die Kanalkodierung mehr Bezug zur Physik des Kanals hat, befindet sie sich im zweiten Fall dichter am Kanal.
- Im letzten Teil der Fälle kann auf die Sicherheitskodierung verzichtet werden.

5.1 zur Einordnung (3)



1) bei Übertragung im Frequenzband oberhalb des Basisbandes
2) bei digitalenSignalen

5.2 Ziele (1)

- Im physikalischen Kanal werden die Signale mit einer gewissen Wahrscheinlichkeit verfälscht. Das Empfangssignal ist nicht identisch zum Sendesignal.
- Mit der Leitungskodierung sollen Übertragungsfehler
  - ..... werden (Error Detection Code) und
  - ..... werden (Error Correction Code - ECC).

5.3 Begriffe, Festlegungen und mathematische Mittel (1)

- Erkennung und / oder Korrektur sind nur möglich, wenn es in den zu sendenden Daten (im Kode) ..... gibt, die bei der Dekodierung ausgenutzt werden können.
- ..... → Abhängigkeiten → Bekanntes → Redundanz
  Beim Hinzufügen von Redundanz muß H sinken

  X: (Ersatz-) Quelle vor der Kanalkodierung
  C: Kanalkode, Ersatzquelle nach der Kanalkodierung)

  $$H(X) > H(C) \quad ? \text{ oder } \quad H_{0\,X} < H_{0\,C}$$

  $$H(X) = P(x_1)\cdot I(x_1) + P(x_2)\cdot I(x_2) + \ldots + P(x_N)\cdot I(x_N)$$
  $$H(C) = P(c_1)\cdot I(c_1) + P(c_2)\cdot I(c_2) + \ldots + P(c_N)\cdot I(c_N)$$

- Da die $P(x_i) = P(c_i)$ sind, müssen die $I(x_i) = I(c_i)$ sein. Lösung:
  - Möglichkeit 1: .....
  - Möglichkeit 2: .....

5.3 Begriffe, Festlegungen und mathematische Mittel (2)

- Kanalkodes:
  - klassisch: Blockkodes
  - neuer: Faltungskodes

  Beispiel für Blockkode ist der ..... (Parity Check):
  Den Kodworten von X mit der Länge M wird jeweils
  ein Kodewort von C mit der Länge N = M+1 zugeordnet über die
  Vorschrift: $c_j = x_j + Paritätsbit$

$$Paritätsbit = \sum_{i=0}^{M-1} x_{j,i} \bmod 2$$

(addiere alle Stellen des Wortes x_j modulo 2)

5.3 Begriffe, Festlegungen und mathematische Mittel (3)

- Ab hier erst einmal anhand linearer Blockkodes betrachtet
  - lineare Blockkodes – Koderate und Redundanzzeichen
    (~Elementarsymbole)
    Datenworte x mit M Elementarsymbolen
    Kodeworte c mit N Elementarsymbolen
    Anzahl Redundanzzeichen = N – M (Beispiel)
    Koderate = M / N (<1)

5.3 Begriffe, Festlegungen und mathematische Mittel (4)

- Modulorechnung
  gilt für diverse Rechenarten (Addition, Multiplikation, Subtraktion, Division, ...)
  (Beispiele)
- Addition von Kodewörtern: stellenweise Addition

$$c_1, c_2, c_3: \text{3 Kodewörter mit N Stellen}$$

$$c_{3,i} = c_{1,i} + c_{2,i} \qquad \text{für alle i=0 bis N-1}$$

- linearer Kode – Definition
  lineare ..... (z. B. Addition) von Kodewörtern ergibt Kodewort
- Vektordarstellung
  Jedes Kodewort der Länge N entspricht einem Vektor F mit N Komponenten
  → alle möglichen binären Kodeworte der Länge N => (Menge) $\boldsymbol{F}_2^N$

5.3 Begriffe, Festlegungen und mathematische Mittel (5)

- Skalarprodukt

$$a, b \in \boldsymbol{F}_2^N$$

$$\langle a, b \rangle = \sum_{i=0}^{N-1} a_i \cdot b_i \ mod\ 2$$

- Hamming-Metrik

  (Metrik: „Zählung“, „Messung“, hier Festlegungen, um etwas meßbar, bewertbar zu machen)

  - Hamming-Gewicht

    Gewicht eines Vektors c ist die Anzahl der von 0 ..... Elemente

$$wt(c) = \sum_{j=0}^{N-1} wt(c_j) \qquad mit \quad wt(c_j) = \begin{cases} 0, & c_j = 0 \\ 1, & c_j \neq 0 \end{cases}$$

  - Hamming-Distanz (.....)

$$dst(c_1, c_2) = \sum_{j=0}^{N-1} wt(c_{1,j} + c_{2,j}) \qquad mit \quad wt(c_j) = \begin{cases} 0, & c_j = 0 \\ 1, & c_j \neq 0 \end{cases}$$

$$dst(c_1, c_2) = wt(c_1 + c_2)$$

5.3 Begriffe, Festlegungen und mathematische Mittel (6)

- Hamming-Metrik (2)
  - Gewichtsverteilung
    W=(w_0, w_1, w_2, ..., w_N) eines Kodes der Länge N gibt an, wieviele Kodewörter w_j mit dem Gewicht j existieren.

$$W(X) = \sum_{j=0}^{N} w_j \cdot x^j \qquad \text{Polynomdarstellung}$$

  - Distanzverteilung
    - Die Distanzverteilung sind die Hamming-Distanzen eines beliebigen Kodewortes zu allen anderen Kodeworten.
    - Die Distanzverteilung eines linearen Kodes ist gleich seiner .....

Beispiel Paritätskode mit N = 4

  - Mindestdistanz (Distanz eines Kodes) und Minimalgewicht

5.3 Begriffe, Festlegungen und mathematische Mittel (7)

- Hamming-Metrik (3)
  - Mindestdistanz (Distanz eines Kodes) und Minimalgewicht

$$d = \min_{\substack{a,c \in \boldsymbol{C} \\ a \neq c}} \{dist(a,c)\}$$

  - für lineare Kodes:

$$Mindestdistanz = minimales\ Gewicht$$

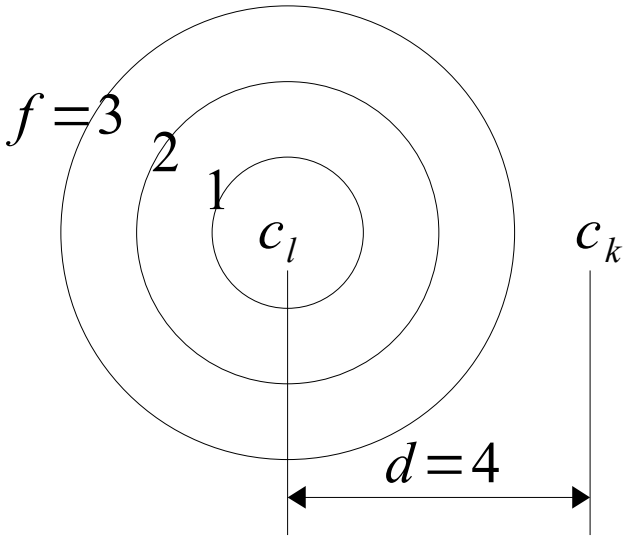
$$d = \min_{\substack{a,c \in \boldsymbol{C} \\ a \neq c}} \{dist(a,c)\} = \min_{\substack{a,c \in \boldsymbol{C} \\ a \neq c}} \{wt(a+c)\} = \min_{\substack{c \in \boldsymbol{C} \\ c \neq 0}} \{wt(c)\}$$

da Gewichtsverteilung = Distanzverteilung
sind auch die Minima gleich

  - Ziel ist die Konstruktion von Kodes mit bestimmter ..... Distanz, die Konstruktion nach ..... ist aber leichter.

5.3 Begriffe, Festlegungen und mathematische Mittel (8)

- Hamming-Metrik (4)
  - Fehlererkennbarkeit
    - Beispielhafte Darstellung der Hamming-Metrik in einer Ebene (nur Modell!)
    - Vektor des Kodewortes bestimmt einen ..... in der Ebene.
    - Abstand zwischen den ..... ergibt sich aus der Hamming-Distanz.
    - Hier werden jetzt nur die beiden ..... mit der geringsten Distanz betrachtet (c_l und c_k).



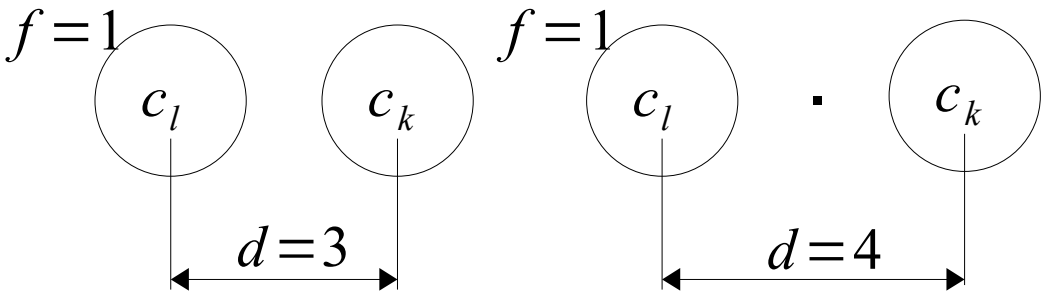
$$c_l, c_k \in \boldsymbol{C}$$

$$Menge\ der\ Vektoren\ \boldsymbol{F} - alle\ Punkte$$

f ist Anzahl der Fehler im empfangenen Kodewort. f ist die Distanz des fehlerfreien Kodeworts zum fehlerbehafteten.
Erkennung aller f < d

5.3 Begriffe, Festlegungen und mathematische Mittel (9)

- Hamming-Metrik (5)
  - Fehlerkorrigierbarkeit
    - Die selbe beispielhafte Darstellung wie bei der Fehlererkennung



    - Fehler sind korrigierbar, wenn der fehlerbehaftete Vektor r (empfangen) eindeutig dem gesendeten Vektor c zugeordnet werden kann. r muß im „Kreis“ mit dem Radius f_{max} liegen.
    - benachbarte „Kreise dürfen sich nicht berühren.
    - Jeder der beiden Vektoren hat den Kreis mit dem Radius f_{max}.

5.3 Begriffe, Festlegungen und mathematische Mittel (10)

- Hamming-Metrik (6)
  - Fehlerkorrigierbarkeit (Fortsetzung)
    - Jeder der beiden Vektoren hat den Kreis mit dem Radius f_{max}.

$$dist(c_l, c_l + f) < dist(c_k, c_l + f)$$

$$wt(f) < wt(c_k + c_l + f)$$

$$wt(f) \leq \left\lfloor \frac{d-1}{2} \right\rfloor \qquad \left\lfloor \frac{d-1}{2} \right\rfloor - \; ganze\ Zahl\ ,\ abgerundet$$

(Beispiel)

5.3 Begriffe, Festlegungen und mathematische Mittel (11)

- Hamming-Metrik (7)
  - Hamming-Schranke
    Wie viele Kodeworte existieren bei gegebener Länge n und Distanz d?
    (Wenn die Länge fest ist und die Mindestdistanz gefordert, wie viele unterschiedliche Kodeworte sind dann zu haben?) (Erklärung)

    Bezug zur Gesamtanzahl der Kodeworte der Länge n?

    dazu: $\boldsymbol{C}(n, k, d)$ – *lin. Blockkode der Länge n, der Dimension k und der Minimaldifferenz d*

    (Struktur) *bei binären Kodes existieren 2^k Worte*

    dazu:
    Untersuchung, wie viele Vektoren es um einen Kodevektor geben kann, die innerhalb einer bestimmten Distanz liegen:

$$Distanz = 1: \binom{n}{1} Vektoren \qquad Distanz = 2: \binom{n}{2} Vektoren$$

$$Distanz = t: \binom{n}{t} Vektoren$$

5.3 Begriffe, Festlegungen und mathematische Mittel (12)

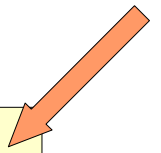
- Hamming-Metrik (8)
  - Hamming-Schranke

$$Distanz = t: \quad \binom{n}{t} \; Vektoren \;\; existieren$$

$$\binom{n}{t} = \frac{n \cdot (n-1) \cdot \ldots \cdot (n-t+1)}{t \cdot (t-1) \cdot \ldots \cdot 1}$$

(Interpretation)

$$2^n \; : \; gesamter \; Vorrat \; an \; möglichen \; Vektoren$$



$$2^k \cdot \left(1 + \binom{n}{1} + \ldots + \binom{n}{e}\right) \leq 2^n \qquad mit \quad e = \left\lfloor \frac{d-1}{2} \right\rfloor$$

in der Klammer jeweils die Gesamtheit der Vektoren um einen Kodevektor herum - liegen in einem „Kreis“ / einer „Sphäre“ um den Kodevektor herum

wenn alle Kodevektoren mit ihren jeweils in der Sphäre darum befindlichen (fehlerhaften) Vektoren den ganzen ..... Vektorraum ausnutzen → perfekter Kode (siehe Beispiele)

5.3 Begriffe, Festlegungen und mathematische Mittel (13)

- Prüfmatrix
  - Mittel zum Prüfen auf .....
  - Matrix H ist so zu konstruieren, daß gilt:

$$H \cdot c^T = 0 \quad \text{(auch } c \cdot H^T = 0 \text{ wird verwendet)}$$

$$((n-k)\,\mathrm{x}\,n)\text{- Matrix}$$

für Paritätskode 3 + 1: $H = (1\,1\,1\,1)$

für Wiederholungskode 1 + 3: $H = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ (Beispiele)

Ist r = c, dann ist s = 0.
Ist s = 0, dann kann r = c sein. Ist s ≠ 0, dann ist r ≠ c.

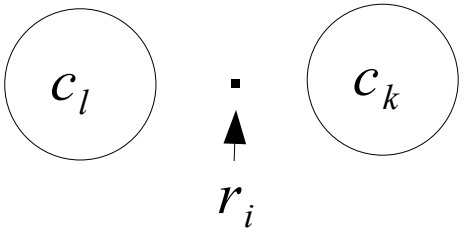
5.3 Begriffe, Festlegungen und mathematische Mittel (14)

- Prüfmatrix - Syndrom
  - Ergebnis der Multiplikation bei einem fehlerhaften (empfangenen) Vektor r
  - Produkt mit (mindestens) einer Zeile der Prüfmatrix zeigt Werte ≠ 0. Dieses Syndrom hängen nur vom Fehler, nicht aber vom dazugehörigen, originalem Kodewort c ab.
  - Wie findet man zu einem festgestellten Syndrom den wahrscheinlichsten Fehler, der genau zu diesem Syndrom s führt?
  - Das ist eine Frage der Dekodierung
    → Dekodierprinzipien

5.3 Begriffe, Festlegungen und mathematische Mittel (15)

- Dekodierprinzipien – mögliche Ergebnisse
  - korrekte Dekodierung: $c_i = c_i'$ $c_i' = r - f$ (empfangenes Wort - Fehler)
  - falsche Dekodierung: $c_i \neq c_i'$ $r_i \in \mathbb{C}$ (empfangenes Wort ist Kodewort)
  - Dekodier.....: r_i nicht zuordenbar

r_i nicht innerhalb einer Korrektur"kugel"



c_i' : *dekodiertes Wort*
c_i : *gesendetes Wort*
r : *empfangenes Wort*
f : *Fehlerwort*

5.3 Begriffe, Festlegungen und mathematische Mittel (16)

- Dekodierprinzipien
  - Fehlererkennung (keine Berichtigung) (siehe: Grenze für Fehlererkennbarkeit) $r = c_i + f$ wenn $f = c_x$, dann falsche Dekodierung kein ..... möglich
  - Maximum Likelihood Detection (ML)
    Auswahl des Kodewortes c_i mit der maximalen .....

    $$P(r|c') \rightarrow max$$

    , bei gleichem P per Zufall
    Die Grenze für die garantierte Korrigierbarkeit wird nicht beachtet. Es kann ja im konkreten Fall Distanzen > d geben.
  - symbolweise Maximum-a-posteriori-Dekodierung (s/s-MAP)
    Entscheidung für jedes Elementarsymbol einzeln.
    r muß kein c' ergeben. Dann liegt ein Dekodierversagen vor.
  - Dekodierung über begrenzte Distanzen
    (siehe nächste Seite)

5.3 Begriffe, Festlegungen und mathematische Mittel (17)

- Dekodierprinzipien (2)
  - Dekodierung über begrenzte Distanzen
    - Begrenzte-Distanz-Dekodierung: $Radius < \left\lfloor \frac{d-1}{2} \right\rfloor$
    - Begrenzte-Mindestdistanz-Dekodierung: (Ausnutzung Mindestkorrigierbarkeit) $Radius = \left\lfloor \frac{d-1}{2} \right\rfloor$
    - Dekodierung über die halbe Mindestdistanz: $Radius = \frac{d}{2}$
  - ML ist eine Dekodierung ..... zwischen konkreten c. (Beispiel)

5.3 Begriffe, Festlegungen und mathematische Mittel (18)

- Dekodierprinzipien (3)

	korrekt	falsch	Versagen	Bemerkung
Fehlererkennung	x	x		keine Korrektur
Maximum Likelihood	x	x		
s/s-MAP	x	x	x	
begrenzte-Distanz-D.	x	x	x	
begrenzte-Mindestd.	x	x	x	
halbe Mindestd.	x	x	x	

(Diskussion)

5.3 Begriffe, Festlegungen und mathematische Mittel (19)

- Dekodierprinzipien - Ansätze zur Dekodierung
  - Tabelle (bzw. Feld) mit allen möglichen Empfangsworten (Empfangsvektoren) r bilden und diese r jeweils dem Kodewort c zuordnen, auf das sie dekodiert werden sollen.
  - Erfordert den ..... des Vektors r mit den Werten in der Tabelle und ermitteln des zugeordneten c.
  - Sicher eingeschränkt bei großer Anzahl von möglichen r und möglichst effizienter Implementierung
  - Methode zum Finden eines Feldes von Vergleichsworten zu r ist unter der Standard-Array-Dekodierung zu finden.
  - Fallweise auch über die Prüfmatrix möglich (siehe systematischer Blockkode unter 5.4)

- 5.3 wird erst einmal verlassen, um unter 5.4 konkrete Verfahren zu betrachten. In 5.4 werden später wieder mathematische Hilfsmittel behandelt.

5.4 einige Kodes und Verfahren (1)

- Blockkode (systematischer)
  - ..... in c steht der Teil x und ..... davon die zugefügte Redundanz.
  - Die Kodierung erfolgt durch ..... des x mit einer Generatormatrix G. Das Ergebnis ist das Kodewort c.

    $$x \cdot G = c$$

    G liefert Erkenntnisse zur Wirksamkeit der Kodierung und unterstützt die Kodekonstruktion.
  - Das empfangene Wort r wird mittels H geprüft. Das Ergebnis ist der Vektor s, das Syndrom.

    $$H \cdot r^T = s$$

    Bei s = 0 ist r = c, genauer gesagt, kann r = c sein.
    Bei s ≠ 0 ist r ≠ c.
  - s dient zur Korrektur. H hat eine spezielle Gestaltung.

5.4 einige Kodes und Verfahren (2)

- Blockkode (systematischer) (2)
  - Erläuterung am Beispiel:

$$\boldsymbol{C}(7,3,3) \;-\; \textit{lin. Blockkode der Länge } 7, \textit{der Dimension } 3$$
$$\textit{und der Minimaldifferenz } 3$$
$$\textit{da binärer Kodes existieren } 2^3 = 8 \textit{ Worte}$$

Die ersten ..... Bit von c entsprechen dem Eingangswert x.
Die jeweils ..... Bit Redundanz werden rechts angefügt.
Die Vorschrift zur Bildung der Redundanz steht in G.

$$G = \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right)$$

jede Zeile ist Kodewort,
jedes weitere Kodewort
durch Linearkombination,
dazu Kodewort 0

entspricht x | für Redundanz

5.4 einige Kodes und Verfahren (3)

- Blockkode (systematischer) (3)
  - Erläuterung am Beispiel:

$$G=\begin{pmatrix}1 & 0 & 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 0 & 1 & 1\end{pmatrix}$$

$$x \cdot G = c$$

(zur Konstruktion und Beispiel)

Die Stellen in x, die nicht 0 sind, bestimmen, welche Zeilen von G in die ..... eingehen.
G hat folgerichtig so viele Zeilen, wie x Stellen hat.
Aus G sind Werte für den Code C ermittelbar:

5.4 einige Kodes und Verfahren (4)

- Blockkode (systematischer) (4)
  - Erläuterung am Beispiel:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Aus G Werte für den Code C ermittelbar:

- Das Minimalgewicht (= Minimaldistanz d) des Kodes ist nicht größer, als
  - das Gewicht jeder Zeile von G und
  - das Gewicht jeder Linearkombination von Zeilen von G.
- Dabei gilt für den linken Teil: Gewicht = Anzahl der Zeilen in der Linearkombination
- $wt_{linker\,Teil} + wt_{rechter\,Teil} = wt$ (am Beispiel)

Das hilft bei der Kodekonstruktion, was später gezeigt wird.

5.4 einige Kodes und Verfahren (5)

- Blockkode (systematischer) (5)
  - Erläuterung am Beispiel:

$$G=\begin{pmatrix}1&0&0&1&1&0&0\\0&1&0&0&1&1&0\\0&0&1&0&0&1&1\end{pmatrix}$$

Aus G Werte für den Code C ermittelbar:

- $wt_{linker\,Teil}+wt_{rechter\,Teil}=wt$

eine Zeile: $wt=1+wt_{rechter\,Teil}$

zwei Zeilen: $wt=2+wt_{rechter\,Teil}$

n Zeilen: $wt=n+wt_{rechter\,Teil}$ (am Beispiel)

5.4 einige Kodes und Verfahren (6)

- Blockkode (systematischer) (6)
  - Erläuterung am Beispiel:

$$G=\begin{pmatrix}1&0&0&1&1&0&0\\0&1&0&0&1&1&0\\0&0&1&0&0&1&1\end{pmatrix}$$

    - Bildung von H

$$H=\left(\begin{array}{ccc|cccc}1&0&0&1&0&0&0\\1&1&0&0&1&0&0\\0&1&1&0&0&1&0\\0&0&1&0&0&0&1\end{array}\right)$$

x

Jede einzelne Stelle der Redundanz wird gegen eine oder die Kombination mehrerer Stellen von x geprüft (Beispiel, Bildung von H).

5.4 einige Kodes und Verfahren (7)

- Blockkode (systematischer) (7)
  - Erläuterung am Beispiel:
    - Anwendung von H

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H \cdot r^T = s^T$$

Die Belegung von s^T gibt über H die Stelle in r an, die mit der höchsten Wahrscheinlichkeit falsch ist. Gleiche Wahrscheinlichkeiten sind möglich. Negation der fehlerhaften Stelle(n) ergibt das richtige Elementarsymbol (da binärer Kode).
s^T kann auch die Linearkombination mehrerer Spalten von H sein.

(Beispiele)

5.4 einige Kodes und Verfahren (8)

- Blockkode (systematischer) (8)
  - Erläuterung am Beispiel:
    - Konstruktion des Kodes - über G (hier Optimierungsversuch am Beispiel)

      Eigentliches Optimierungsziel ist aber die Vergrößerung der minimalen Distanz d.

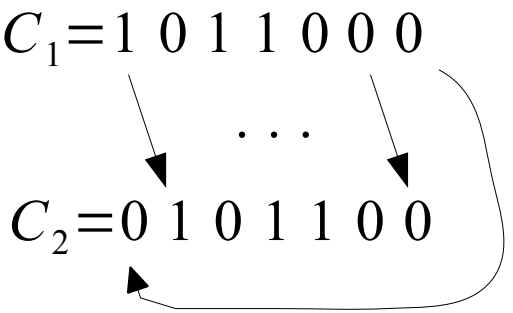
      Optimiert wird jedoch praktisch auf eine Erhöhung des minimalen ....., da die ..... leichter überschaubar ist, als die Distanzverteilung.

      Die Erkenntnisse von einigen Seiten zuvor kommen zur Anwendung.

5.4 einige Kodes und Verfahren (9)

- zyklischer Kode
  - Blockkode, bedeutet .....
  - hier linearer Blockkode, bedeutet .....
  - hier systematischer, linearer Blockkode, bedeutet .....
  - zyklischer, ... Blockkode
    - das bedeutet, ... **und** ..... eines Kodewortes ergibt wieder ein (gültiges) Kodewort

      Beispiel:



    - zum Vorteil dieser Eigenschaft später

5.4 einige Kodes und Verfahren (10)

- Hilfe aus der abstrakten Algebra
  - **Körper**: ausgezeichnete algebraische Struktur, bei der Addition, Subtraktion, Multiplikation und Division wie bei reellen Zahlen ausführbar sind (Bspl.)
  - **Galois-Feld - GF**: Körper mit endlich vielen Elementen $q = p^n$
    p ist Primzahl, n ist natürliche Zahl, Berechnungen modulo p
    (Referenzen zu binären Blockkodes)
  - **zyklischer Körper**: bestimmte GF: alle Elemente von GF(p) \ {0} erzeugbar mittels Potenzen eines (bestimmten) Elements
  - **primitives Element**: das Element eines zyklischen Körpers, dessen Potenzen alle Elemente des GF(p) erzeugt (ist nicht weiter zerlegbar)

    Beispiel:
    GF (5^1) - ganze Zahlen 0...4, d. h., <5; 0 wird gesondert behandelt
    p=5, n=1, q=5

    2 ist primitives Element: $2^1 = .., \; 2^2 = .., \; 2^3 = .., \; 2^4 = .. \quad alles \; mod \; 5$

    3 ist primitives Element: $3^1 = .., \; 3^2 = .., \; 3^3 = .., \; 3^4 = .. \quad alles \; mod \; 5$

5.4 einige Kodes und Verfahren (11)

- zyklischer Kode – Hilfe aus der abstrakten Algebra (2)
  - **GF**, **zyklischer Körper** und **primitives Element** als Grundlage für hier behandelte zyklische Kodes
  - Elemente: Kodeworte
  - Darstellung der Kodeworte: ..... mit binären Faktoren (vorher Vektor mit binären Werten) → Elemente des Körpers sind .....

$$P_i = g_m \cdot x^m + g_{m-1} \cdot x^{(m-1)} + g_{m-2} \cdot x^{(m-2)} + \ldots + g_1 \cdot x^1 + g_0 \cdot x^0$$

$$g_i \in \{0, 1\}$$

Beispiel: $c_i = 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0$ bisher

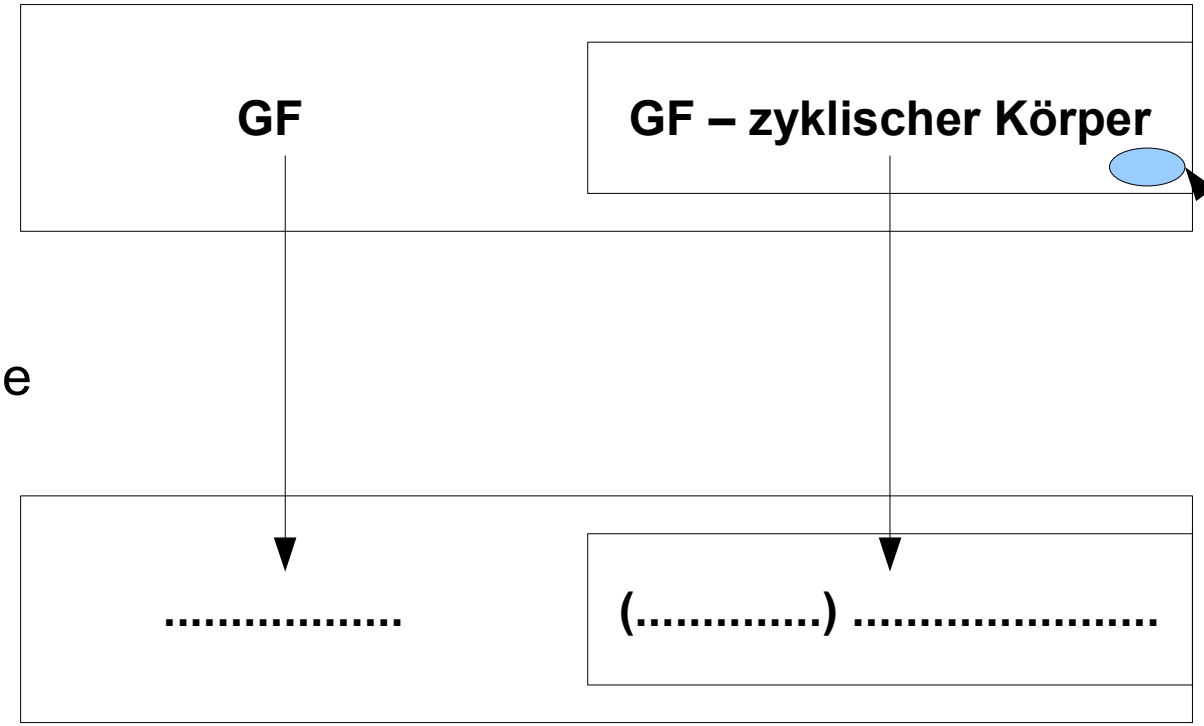
$$P_i = 0 \cdot x^6 + 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 0 \cdot x^0$$ neu

- Modulogrenze: $x^p + 1$ begrenzt den Körper

im Beispiel: $x^7 + 1$

5.4 einige Kodes und Verfahren (12)

- zyklischer Kode – Hilfe aus der abstrakten Algebra (3)
  - Ein zyklischer Körper kann Teilkörper aus einem GF sein.

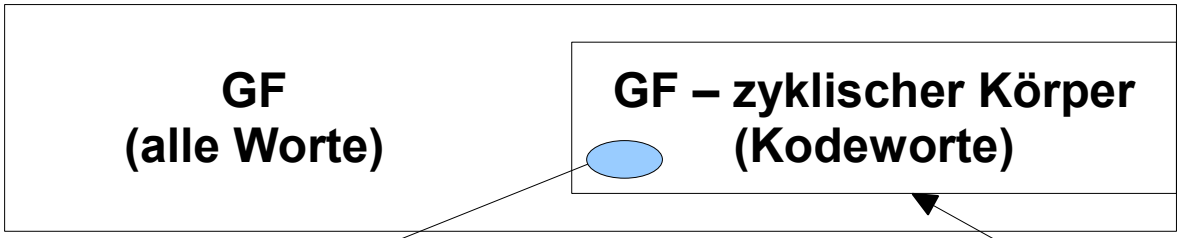


  - Kode

z. B. ..... mit 7 Bit ..... mit 7 Bit

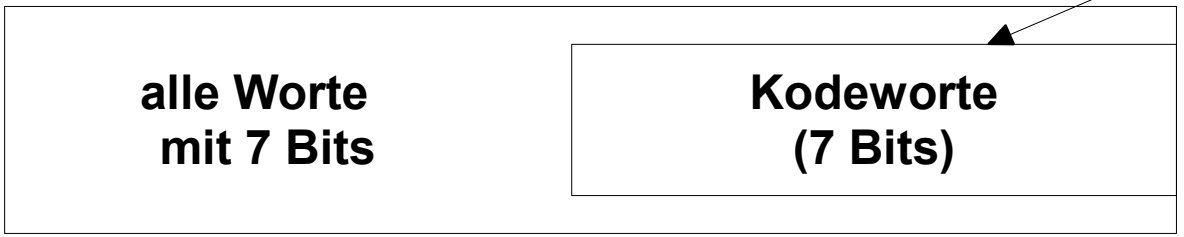
5.4 einige Kodes und Verfahren (13)

- zyklischer Kode – Hilfe aus der abstrakten Algebra (4)
  - Ein zyklischer Körper kann Teilkörper aus einem GF sein.



..... Element (Polynom) als Basis * alle Polynome aus den ..... → alle Elemente (Polynome) des zyklischen Körpers, also alle .....

Beispiel: ..... mit 3 Bits → ..... mit 7 Bits



5.4 einige Kodes und Verfahren (14)

- zyklischer Kode – Hilfe aus der abstrakten Algebra (5)

primitives Element (Polynom) als Basis * alle Polynome aus den Datenworten → alle Elemente (Polynome) des zyklischen Körpers, also alle Kodeworte

Wenn alle Kodewortpolynome durch die Multiplikation der Datenwortpolynome mit dem primitiven Polynom gebildet werden, so lassen sich alle Kodewortpolynome ohne Rest durch das primitive Polynom teilen!

Das primitive Polynom ist Generatorpolynom.

5.4 einige Kodes und Verfahren (15)

- zyklischer Kode – Hilfe aus der abstrakten Algebra (6)

primitives Element (Polynom) als Basis * alle Polynome aus den Datenworten → alle Elemente (Polynome) des zyklischen Körpers, also alle Kodeworte

analoges Verfahren: Division anstelle der Multiplikation (für schaltungstechnische Realisierung)

- zyklischer Kode – hier über Divisionsverfahren

alle Polynome aus den Datenworten → potenziert auf Grad von **C** : primitives Element (Generatorpolynom) → alle Elemente (Polynome) des zyklischen Körpers, also alle Kodeworte

5.4 einige Kodes und Verfahren (16)

- zyklischer Kode – hier über Divisionsverfahren (2)
  - Eingangswert bekannter Wert

$$\text{Datenwort} \cdot x^m \;:\; \text{Generatorpolynom} = \text{ganzes Polynom} + \text{Restpolynom}$$

[Zahl 1] * : [Zahl 2] * = [ganzer Teil]* + [Rest]*

hier uninteressant Redundanz

Datenwort - Stützstellenwerte des Polynoms	+	**Divisionsrest - Stützstellenwerte des Polynoms**	=	**Kodewort - Stützstellenwerte des Polynoms**

Die Division ist die Methode, zu den ..... zu kommen.
Die Kodeworte gehören einem ..... Körper an, denn sie sind durch das Generatorpolynom teilbar. Das ergibt sich praktisch durch das Anfügen des .....

[in Klammern - Analogie]*

5.4 einige Kodes und Verfahren (17)

- zyklischer Kode – hier über Divisionsverfahren (3)
  - Beispiele, konkrete Wert:

(N,K)-Kode	(7,3)	(7,4)
• Redundanzbits: N-K	4	3
• Grad Divis.-Rest: N-K-1	3	2
	$g_3 x^3+g_2 x^2+g_1 x+g_0 1$	$g_2 x^2+g_1 x+g_0 1$
• Grad Generatorpolynom: N-K-1+1 (muß um 1 größer sein, als Grad Rest)	4	3
	$x^4+g_3 x^3+g_2 x^2+g_1 x+1$	$x^3+g_2 x^2+g_1 x+1$
• Grad Datenpolynom: K-1	2	3
	$g_2 x^2+g_1 x+1$	$g_3 x^3+g_2 x^2+g_1 x+1$
• Erweiterung auf Grad des Polynoms, daß Elemente von **C** repräsentiert	6	6
Erweitern um: N-1-(K-1)	4	3

Achtung!

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

5.4 einige Kodes und Verfahren (18)

- zyklischer Kode – hier über Divisionsverfahren (4)
  - Ermittlung der Generatormatrix G
    Diese muß K unabhängige Zeilen enthalten.

Bspl.: $g_{31}(x) = x^3 + x + 1$ (7,4)-Kode: 4 Datenbits, 3 Redundanzbits
→ 4 Zeilen in G

$$G = \left(\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right) \qquad \begin{array}{l} x^3 \cdot g(x)\, Mod\,(x^7+1) \\ x^2 \cdot g(x)\, Mod\,(x^7+1) \\ x \cdot g(x)\, Mod\,(x^7+1) \\ g(x) \end{array}$$

Datenbits | Redundanzbits

$$G = \left(\begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right) \qquad \begin{array}{l} Z_{1\,alt} + Z_{3\,alt} + Z_{4\,alt} \\ Z_{2\,alt} + Z_{4\,alt} \\ Z_{3\,alt} \\ Z_{4\,alt} \end{array}$$

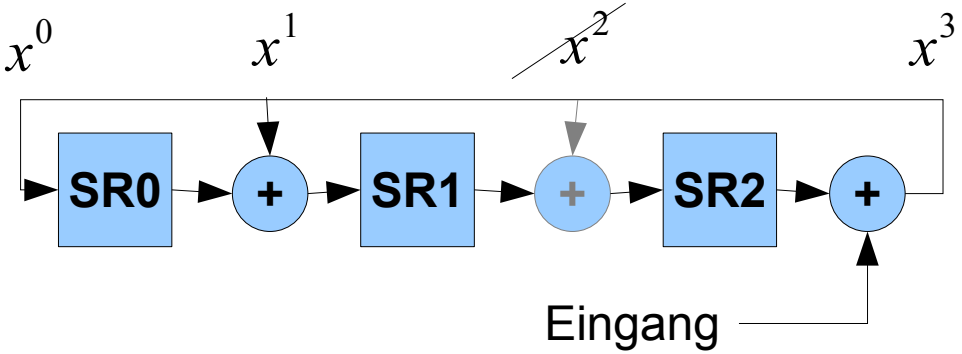
systematische Form durch Linear-kombination

5.4 einige Kodes und Verfahren (19)

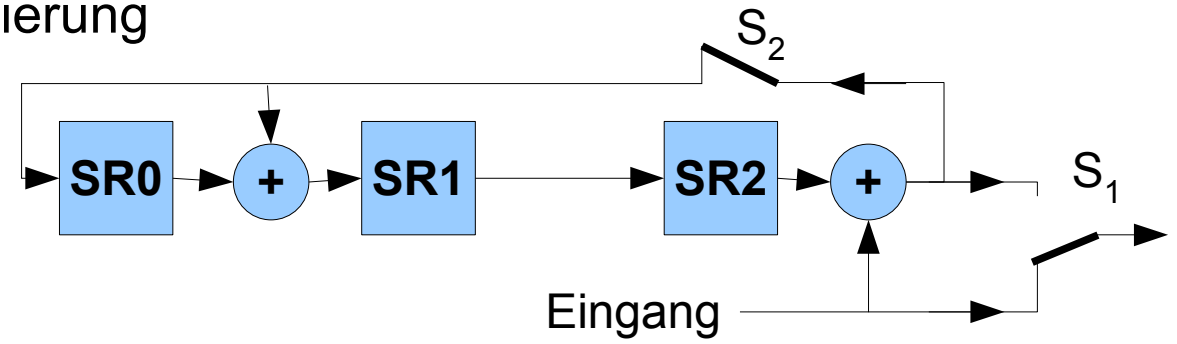
- zyklischer Kode – hier über Divisionsverfahren (5)
  - Abbildung in Hardware (oder Software) über rückgekoppelte Schieberegister (SR)

$$g_{31}(x) = x^3 + x + 1$$ Grad=3 → 4 Werte, 3 davon im SR gespeichert

Abbildung Polynom



Kodierung

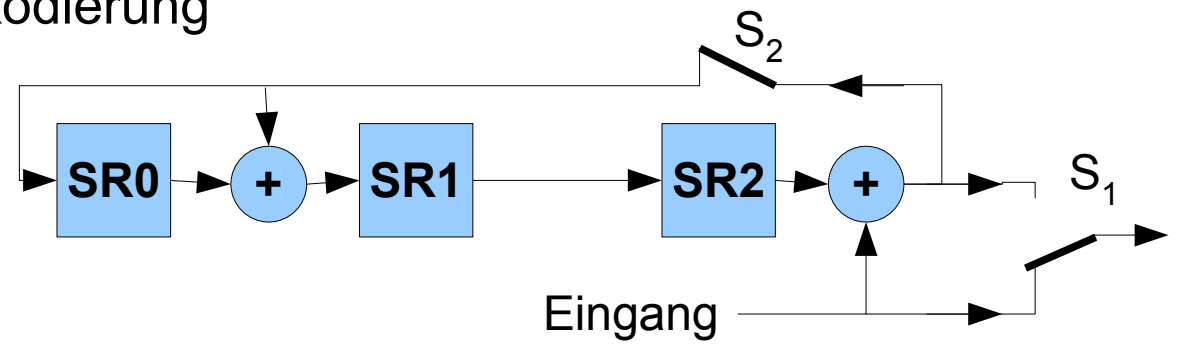


K Bit (Datenbits):
S_1 unten, S_2 zu

N-K Bit (Redundanz):
S_1 oben, S_2 auf

5.4 einige Kodes und Verfahren (20)

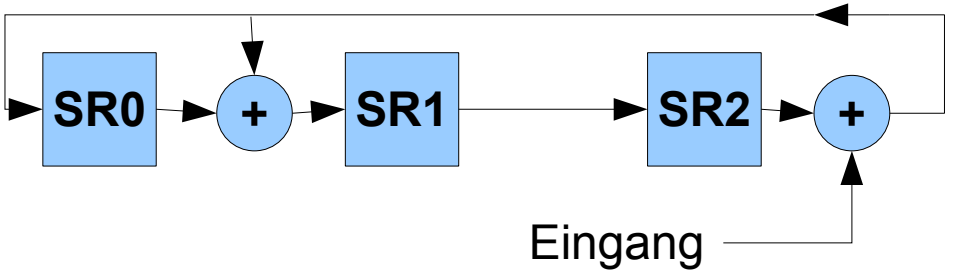
- zyklischer Kode – hier über Divisionsverfahren (6)
  - Abbildung in Hardware (oder Software) über rückgekoppelte Schieberegister (SR)
    Kodierung



K Bit (Datenbits):
S_1 unten, S_2 zu

N-K Bit (Redundanz):
S_1 oben, S_2 auf

Dekodierung - .....



zu Beginn SR auf Startwert (im Basisverfahren auf 0)

nach ganzem Kodewort SR auf Standardwert (im Basisverfahren auf 0)

5.4 einige Kodes und Verfahren (21)

- zyklischer Kode – hier über Divisionsverfahren (7)
  - Abbildung in Hardware (oder Software) über rückgekoppelte Schieberegister (SR)
    Dekodierung - .....
    - Die ..... sind durch das ..... ohne Rest teilbar: „nach ganzem Kodewort steht das SR auf dem Standardwert (im Basisverfahren auf 0)“
    - Wenn ein empfangenes Wort nicht ohne Rest teilbar ist, liegt ein Fehler vor: das SR enthält .....
    - fehlerhafte Worte können als Addition des Kodewort-Polynoms und eines Fehlerpolynoms aufgefasst werden. Nur wenn dieses Fehlerpolynom auch ohne Rest durch das Generatorpolynom teilbar ist, wird der Fehler nicht erkannt, liegt also eine ..... vor.

5.4 einige Kodes und Verfahren (22)

- zyklischer Kode – hier über Divisionsverfahren (8)
  - Abbildung in Hardware (oder Software) über rückgekoppelte Schieberegister (SR)
    Dekodierung - .....
    - Eine Fehlerkorrektur ist so erklärbar:
      - Jedes Datenwort wird als Werte eins Polynoms an K festgelegten ..... betrachtet. Der Verlust oder die Verfälschung des Wertes einer ..... wäre nicht korrigierbar.
      - Bei der Bildung der Kodeworte werden redundante ..... zugefügt, die selber Werte beisteuern. Beim Verlust oder der Verfälschung von Werten an ..... kann ggf. das originale Datenwort rekonstruiert werden.
        Über geeignete mathematische Verfahren kann das ursprüngliche Datenwort ermittelt werden.
        Das könnte man im weiteren Sinne analog zu der Approximation einer Kurve betrachten.

5.4 einige Kodes und Verfahren (23)

- Reed-Solomon-Kode, RS
  - leistungsfähige Kanalkodierung
  - Anwendung z. B. bei DVB, DAB, Audio-CD, Raumfahrtkommunikation
  - arbeitet mit Polynomen
  - Die Polynomkoeffizienten sind Elemente eines Körpers. Sinnvollerweise werden endliche Körper verwendet.
    Das ist nicht der Körper, dessen Elemente die Polynome sind!
  - RS-Kodes sind zyklische Kodes.
    Die zuvor behandelten binären zyklischen Blockkodes sind eine Untermenge der RS-Kodes. Die Polynomkoeffizienten sind Elemente des binären Körpers: {0;1}.
  - Verfahren für die RS-Kodes, speziell zur Dekodierung, sind für die zyklischen binären Kodes verwendbar.
  - weiterführend:
    http://www-math.upb.de/~mathkit/Inhalte/MatheCD/i30.html

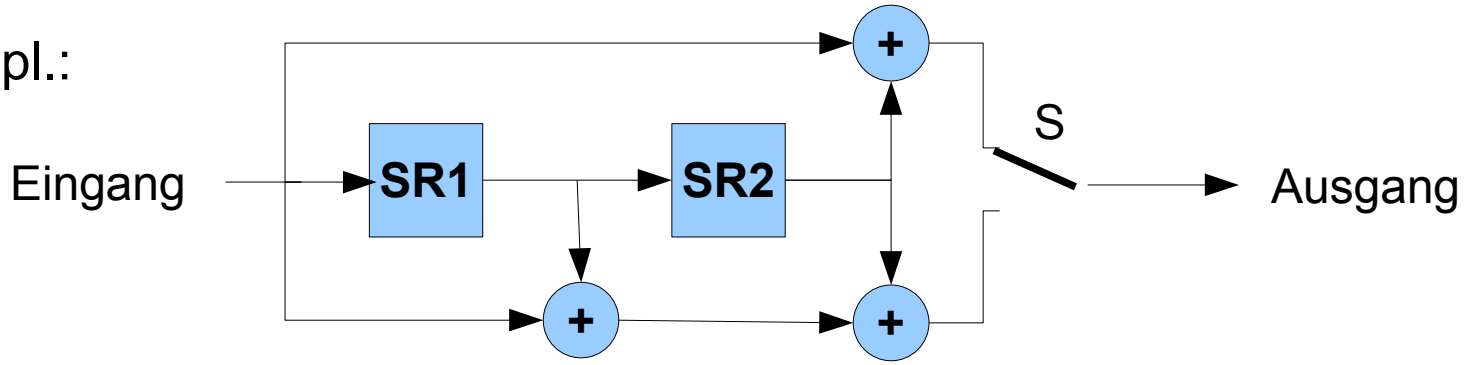
5.4 einige Kodes und Verfahren (24)

- BCH-Kodes, BCH für Bose-Chaudhuri-Hocquenghem
  - Anwendung z. B. bei GSM
  - Definiert über Körpern
  - RS-Kodes und BCH-Kodes überschneiden sich
  - Die weiter vorn beschriebenen binären zyklischen Kodes gehören zu den BCH-Kodes (spezielle Generatorpolynome).

5.4 einige Kodes und Verfahren (25)

- Faltungskodes, convolutional codes
  - Ergebnis der Kodierung von Daten ist auch abhängig von dem Ergebnis vorangegangener Kodierungen
    Quasi Verteilung der Information einer Stelle der unkodierten Daten auf mehrere Stellen der kodierten Daten
  - Anwendung des Verfahrens der Faltung
  - Anwendungen: z. B. Mobilfunk, WLAN, Datenspeicherung
  - im binären Bereich mittels Schieberegister und XOR realisierbar,
    rekursive und nicht rekursive Strukturen

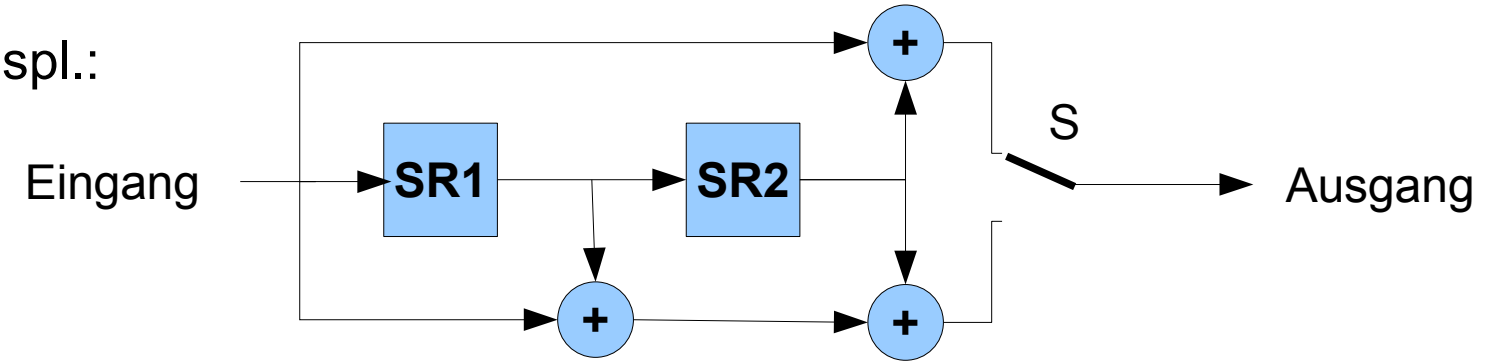
Bspl.:



  - Leistungsfähige Dekodierferfahren, z. B. Viterbi-Algorithmus (Trellis),
    maximum likelihood

5.4 einige Kodes und Verfahren (26)

- Faltungskodes, convolutional codes
  - Bspl.:



Constrained length (Constrained Länge) = 3 - Tiefe Gedächtnis

Eingangsdaten: 1 1 0 1 0 0 1 | 0 0

Daten – Bits | Tail – Bits

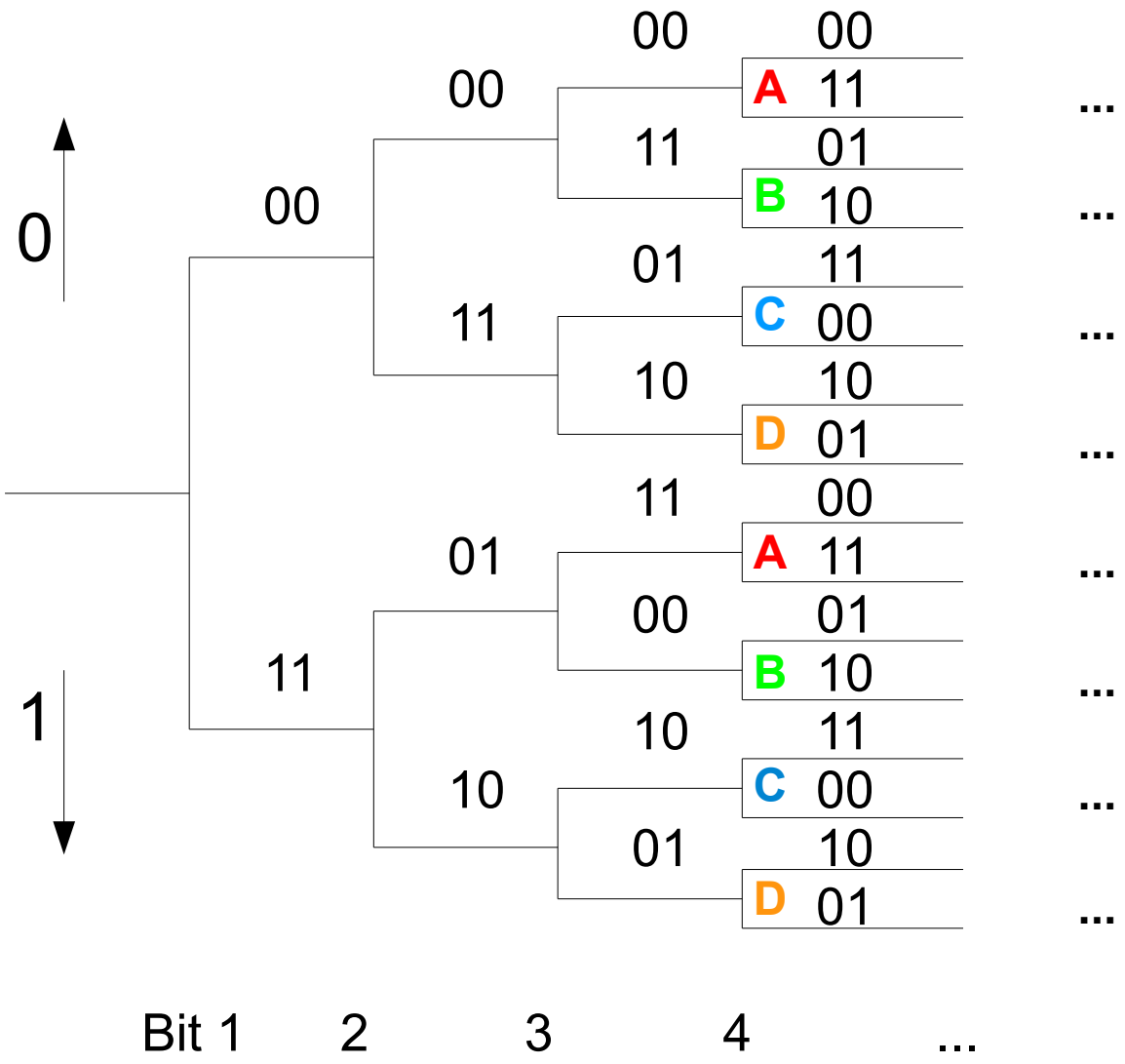
Zustand: .....

Ausgangsdaten: .....

Koderate: $R_C = \frac{\quad}{\quad} = -$ Ziel: .....

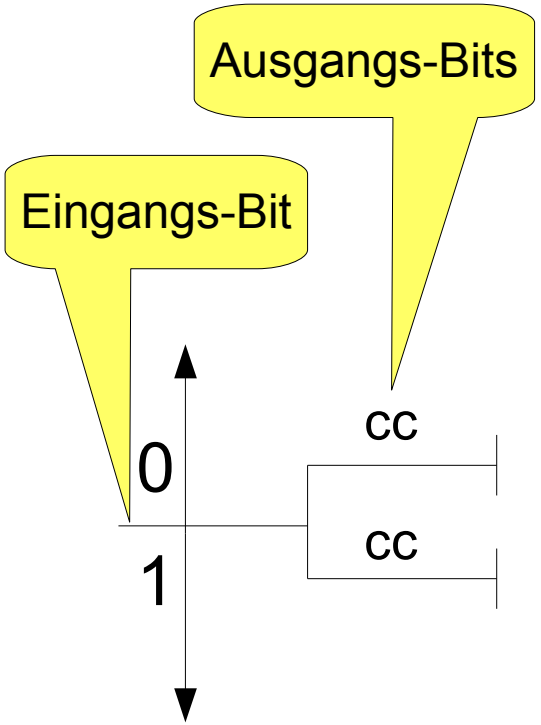
5.4 einige Kodes und Verfahren (27)

- Kodebaum zum Bspl.:



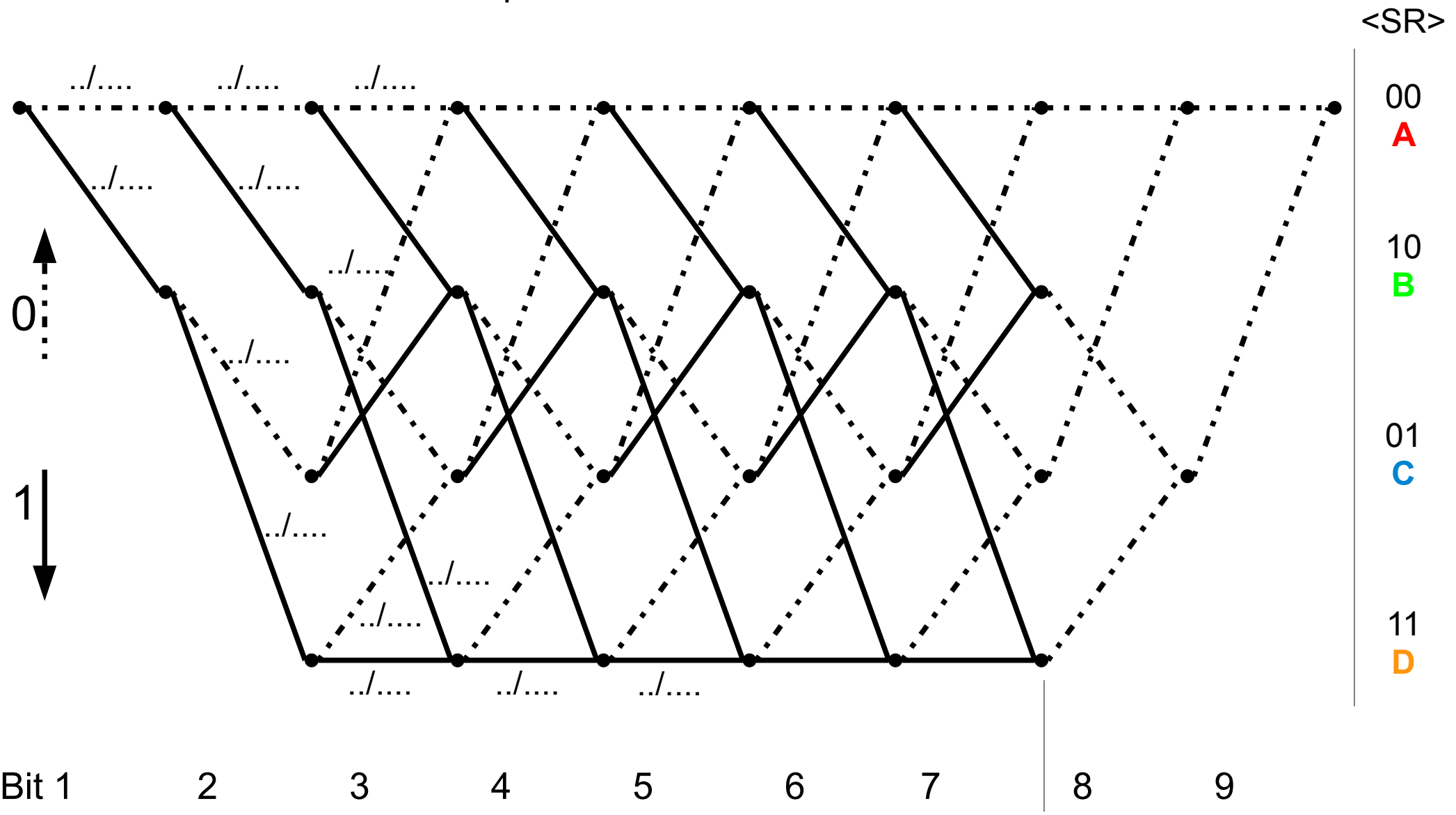
<SR1> <SR2>

A	0	0
B	1	0
C	0	1
D	1	1



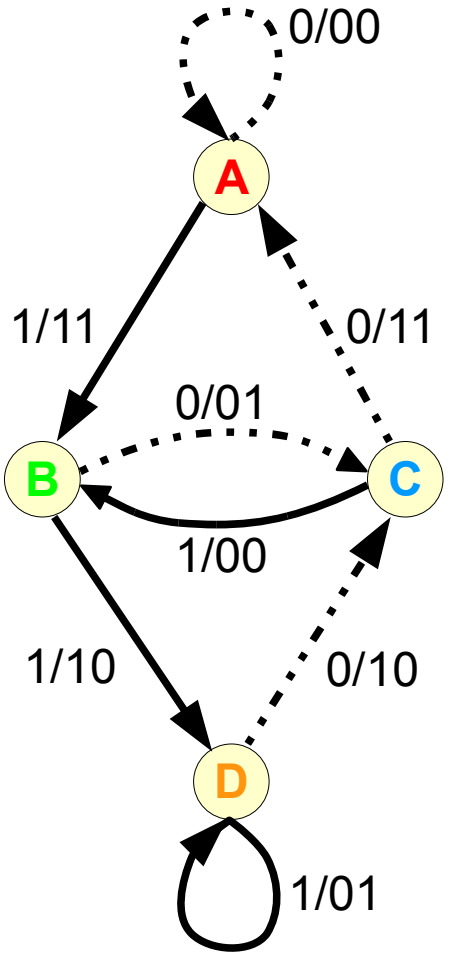
5.4 einige Kodes und Verfahren (28)

- Trellis zum Bspl.:



5.4 einige Kodes und Verfahren (29)

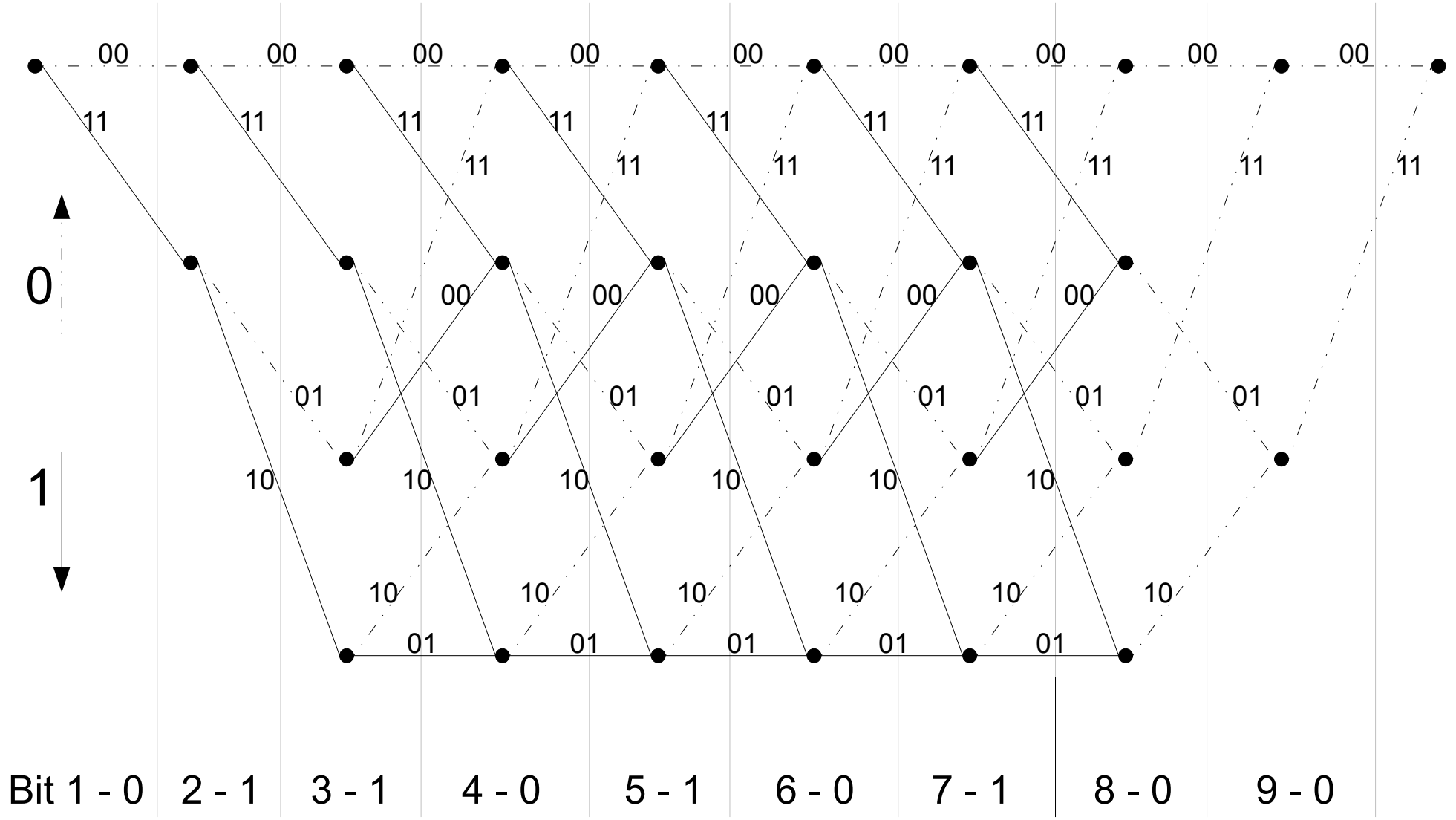
- Zustandsdiagramm zum Bspl.:



Ziel: großer Unterschied der Ausgangsdaten bei jeder Entscheidung

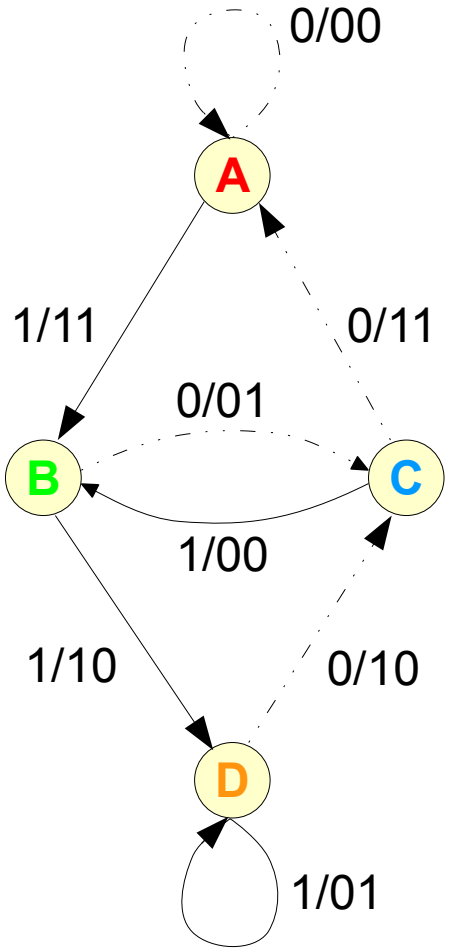
5.4 einige Kodes und Verfahren (30)

- Trellis zum Bspl. - Kodierung der Folge 0110101.00:



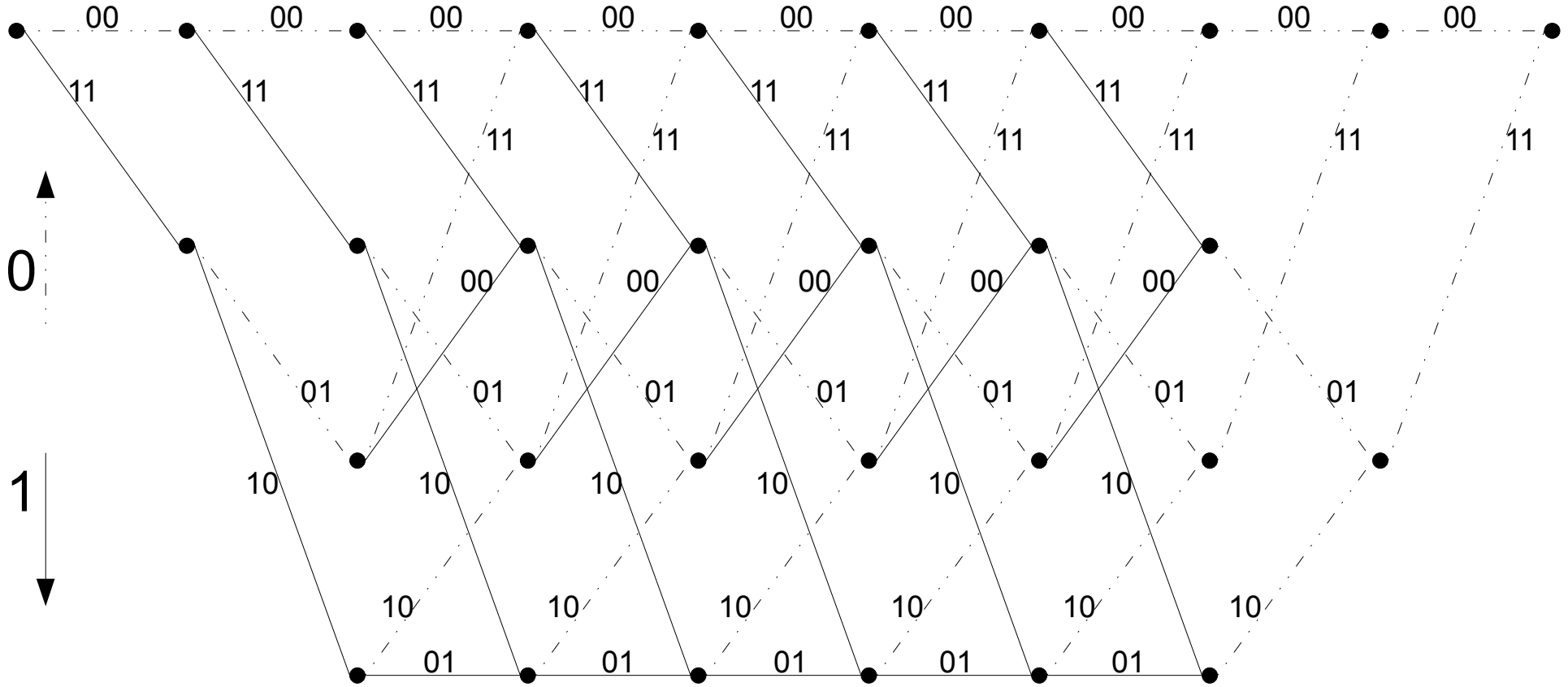
5.4 einige Kodes und Verfahren (31)

- Dekodierung zum Bspl. - 00 11 10 10 10 01 00 . 01 11
  Bezug auf das Zustandsdiagramm:



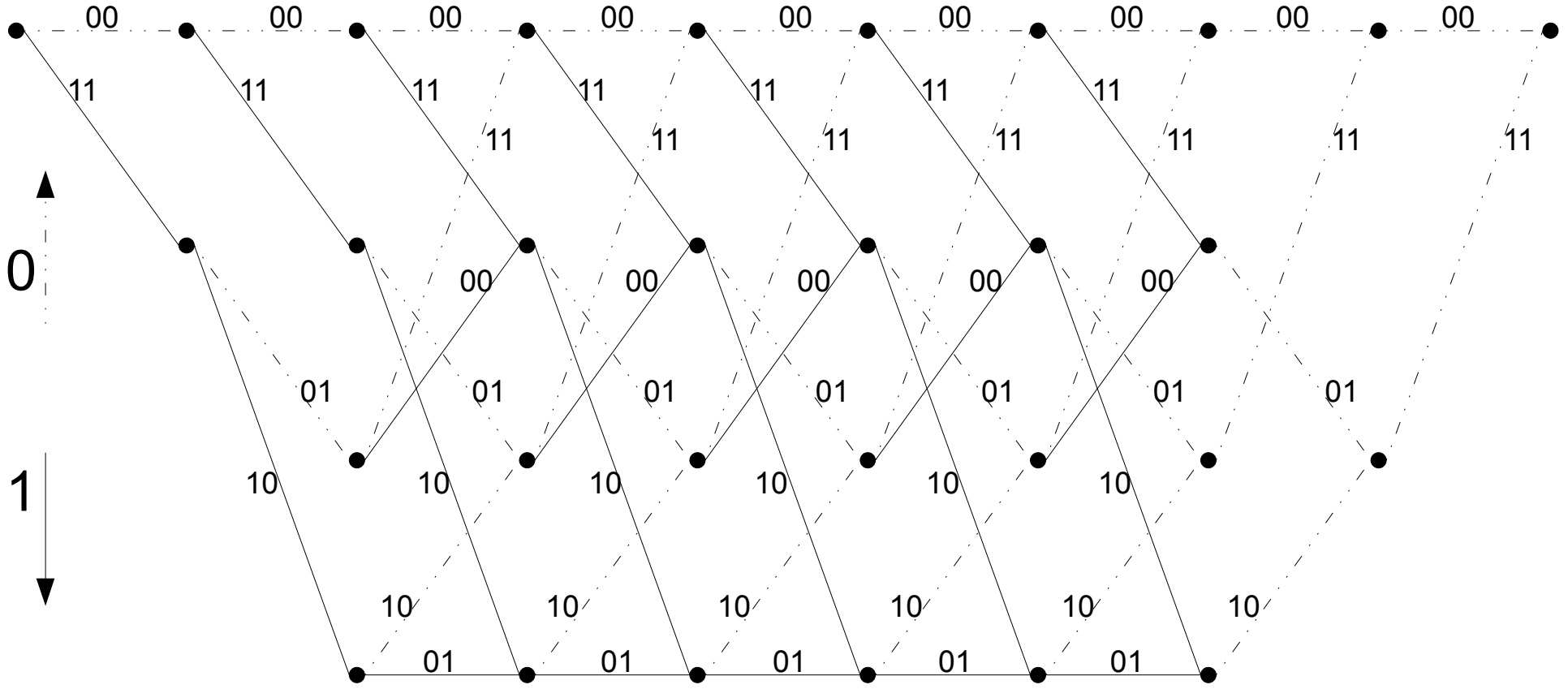
5.4 einige Kodes und Verfahren (32)

- Dekodierung zum Bspl. - 00 11 10 10 10 01 00 . 01 11
  Viterbi-Algorithmus (siehe auch Zustandsdiagramm):



5.4 einige Kodes und Verfahren (33)

- freie Distanz - zum Bspl.:



$$d_f = \min_{v \neq 0} wt(v)$$

Gewicht dieses Pfades – min. Abstand zum 0-Pfad

5.4 einige Kodes und Verfahren (34)

- freie Distanz.:

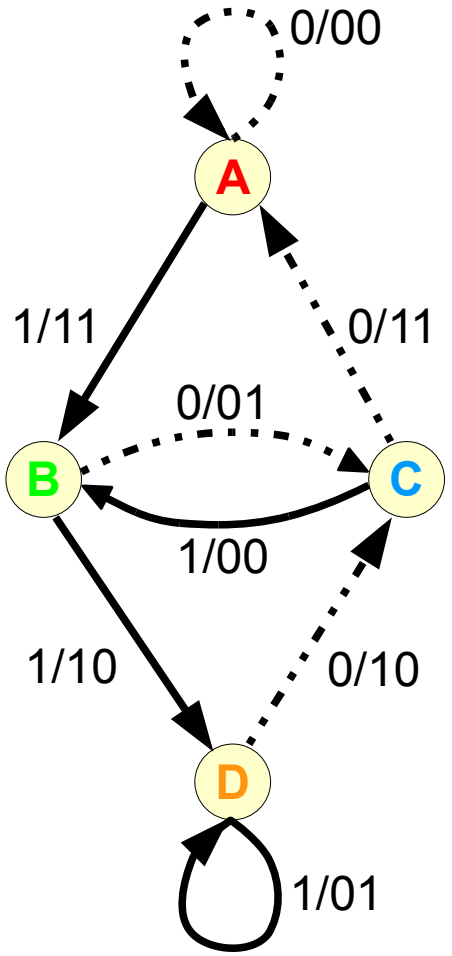
  min. Abstand zum 0-Pfad – Gewicht dieses Pfades

  Pfad unendlich lang! -???

  - nur Sequenzen, die bei t=0 den Nullpfad verlassen,
    denn andere Sequenzen sind nur verzögerte dazu
  - nur Sequenzen, die im Nullpfad enden,
    denn min Gewicht ohne zusätzliche Schleifen, da diese das Gewicht erhöhen
  - keine Sequenzen, die den Nullpfad mehrfach verlassen,
    denn Weg bis zum ersten Erreichen der Nullsequenz bereits eine Sequenz mit geringerem Gewicht ist
  - keine Sequenzen, die nach $t>2^m$ den Nullpfad noch nicht erreicht haben, denn dafür ist / sind Schleife/n nötig

5.4 einige Kodes und Verfahren (35)

- freie Distanz zum Bspl.:



d_f ist Kodeeigenschaft, d. h.,

unabhängig von der Zuordnung der Eingangsdaten zu den Kodeblöcken

5.4 einige Kodes und Verfahren (36)

- Hard decision

- Soft Decision

  3 Bit Auflösung → entspricht + 2 dB SNR

5.4 einige Kodes und Verfahren (37)

- Erasure

- Punktierung

5.4 einige Kodes und Verfahren (38)

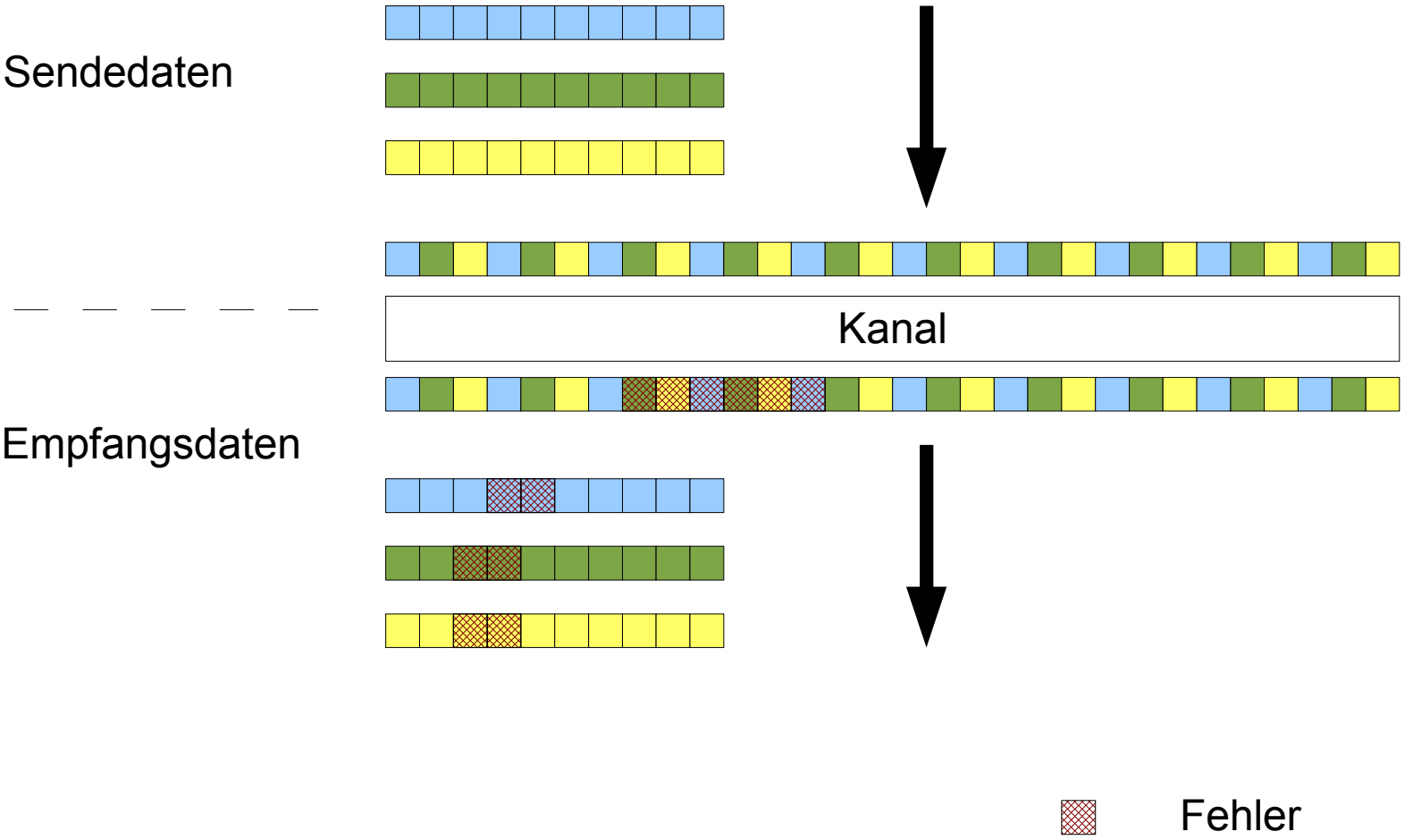
- Kodeverkettung
  - Verkettung mehrerer Kanalkodierungen
  - Kombination der jeweiligen Eigenschaften
  - bessere Anpassung an Erfordernisse
  - Anwendung z. B. bei der Datenübertragung in der Raumfahrt (Empfehlung CCSDS100.0-G-1 Telemetry: RS und Faltungskode)
  - Teilweise noch junges Gebiet (Verkettung von Faltungskodes)– was kommt noch?

5.4 einige Kodes und Verfahren (39)

- Kodeverkettung (2)
  - Eine interessante Variante ist die Kombination von Kodes und Interleaving. (genau genommen keine Kodeverkettung, da Interleaving nicht zur Kodierung gehört)
  - Problem: Fehler im Kanal treten oft als ..... auf. Im Mittel ist die Fehlerdichte zwar so, daß eine bestimmte Kanalkodierung alle Fehler korrigieren kann, die ..... überfordert den verwendeten Kode jedoch zeitweise.
  - Lösung: Es wird mit Interleaving gearbeitet. Mehrere Codeworte (Blöcke) werden ineinander verschachtelt, z. B. bitweise. Ein ..... wirkt nun auf mehrere Blöcke oder, anders gesehen, der ..... wird auf mehrere Blöcke verteilt. Die verwendete Kanalkodierung kommt dann mit der geringeren Anzahl von Fehlern in den einzelnen Blöcken wieder zurecht.

5.4 einige Kodes und Verfahren (40)

- Interleaving (Kodierung? Keine Redundanz!)



5.4 einige Kodes und Verfahren (41)

- Weiteres Material z. B. in:
  - M. Bossert, Kanalcodierung, B. G. Teubner, Stuttgart 1998
  - und sowieso im Internet