UNIVERSITÄT DUISBURG ESSEN

Error correction, - detection and cyphers

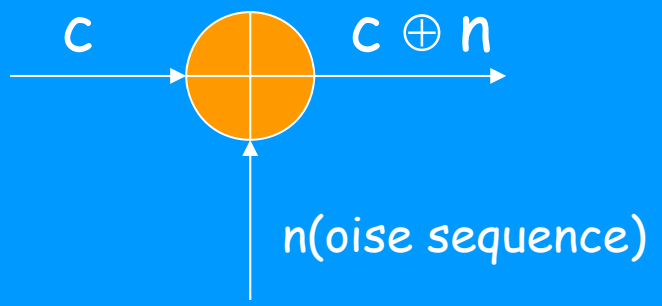
A.J. Han Vinck
University of Duisburg/Essen
June 29, 2013
Vinck@iem.uni-due.de

content

- Influence of transmission errors

A model for transmission errors

We assume binary transmission, errors occur as inversion of data at the receiver



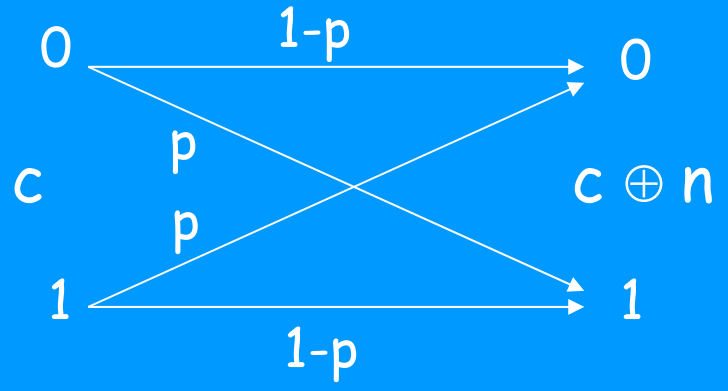
Example: c = 0 0 1 1; n = 0 0 1 0

then: c ⊕ n = 0 0 0 1

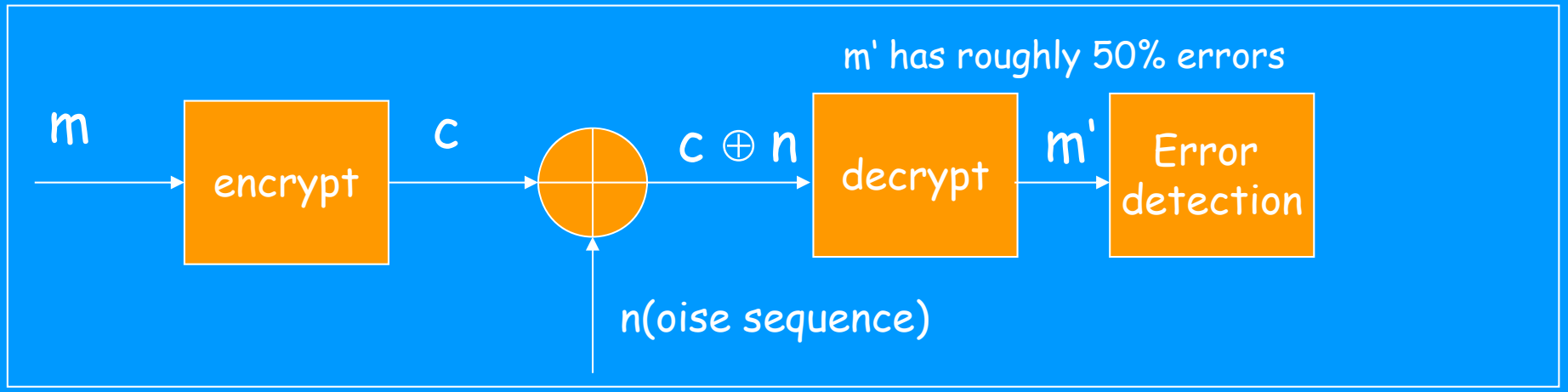
In a probabilistic desciption:

Probability (n = 1) = 1 – Prob(n = 0) = p

the binary symmetric channel model (BSC)



transmission errors destroy the crypto-system



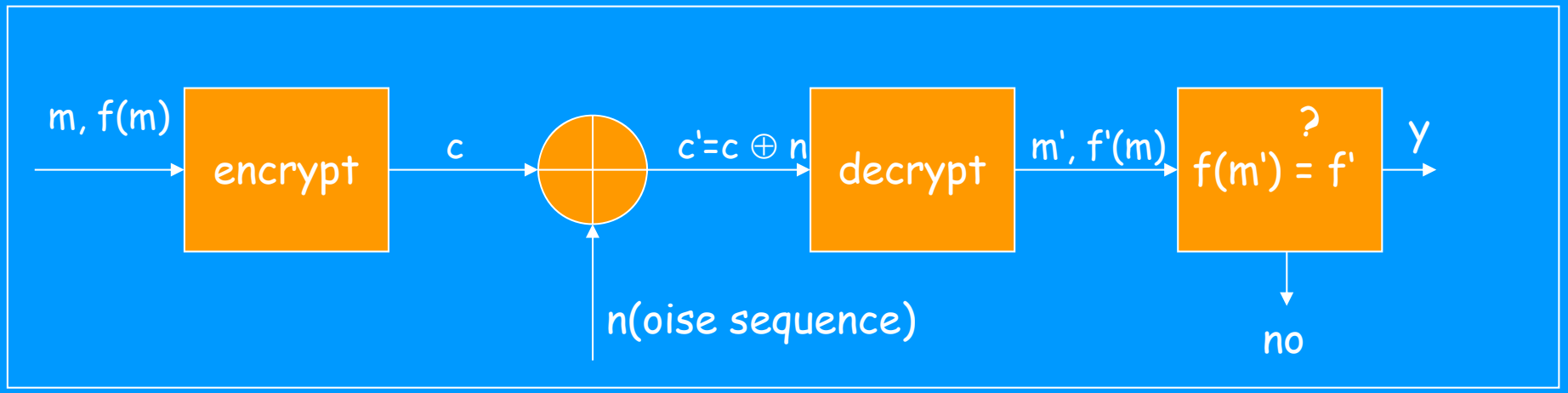
Note:

Error detection can be based on the property that m is a redundant input, like text.

However, this is not a good input to any crypto-system to obtain high security.

Q. What to do when m is a „random" type of input?

We use redundancy to detect errors



Note: f has to be specified, but assume it depends on all information digits in some way

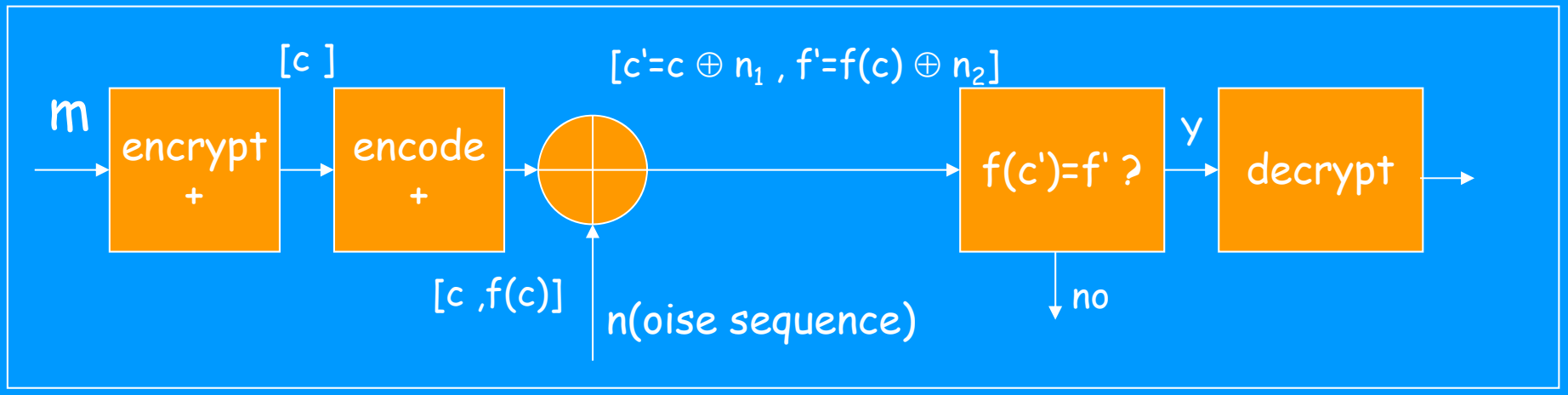
Then:

with high probability, errors will be detected since m' and f' each have about 50% errors.

However, f(m) introduces (additional) redundancy into the encryption input!

Remark: Change f(m) into an all zero string. Give your comment on the idea?

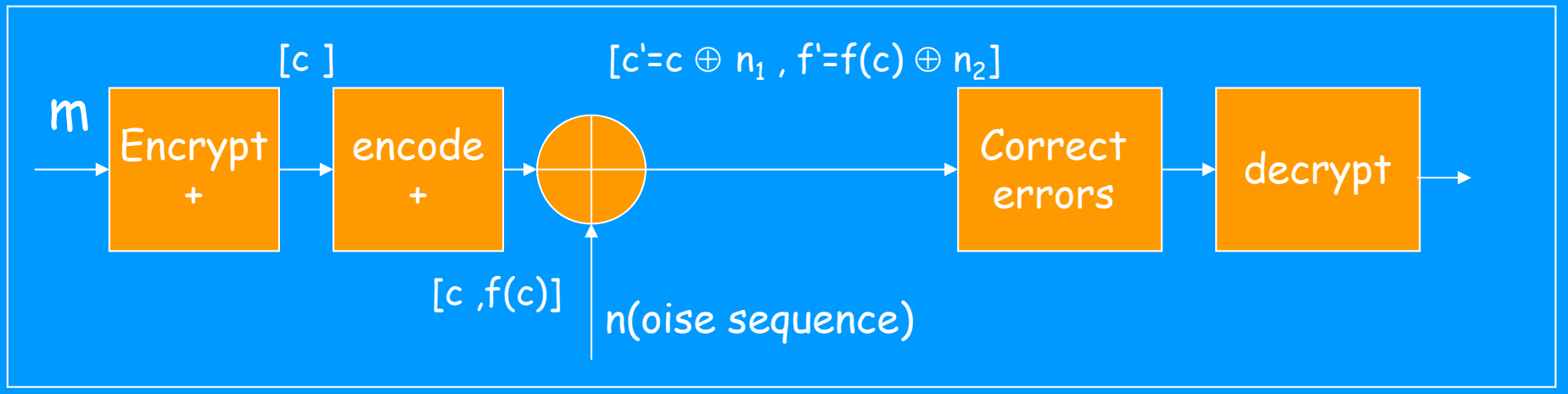
Make redundancy dependent on the ciphertext



Note: f' can be used to detect errors in c' .

Depending on the redundancy, this can be very efficient and with low undetected error probability

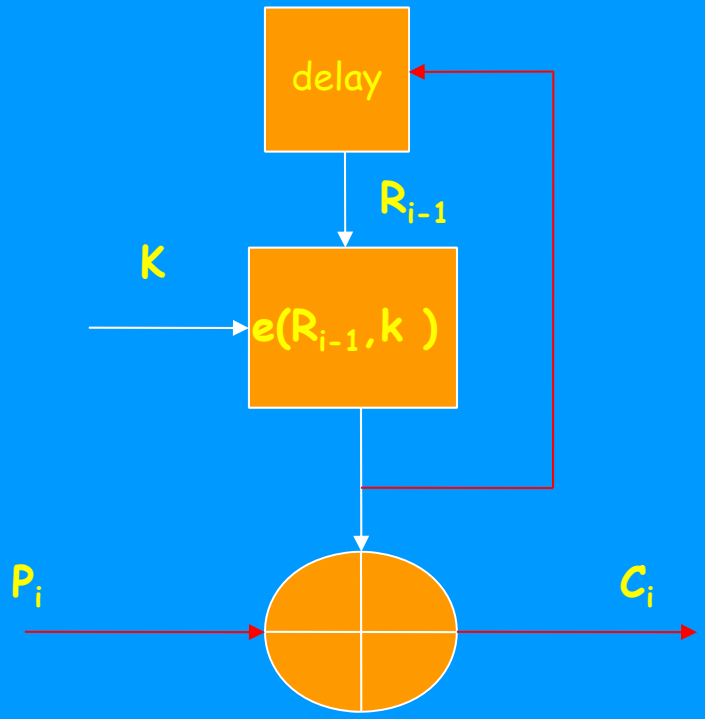
Errors can also be corrected



Note: f' can be used to correct errors in c' .

Depending on the redundancy, this can be very efficient and with low decoding error probability

No error propagation in Output Feedback Mode



$C_i = P_i \oplus e(R_{i-1}, k)$

$P_i = C_i \oplus e(R_{i-1}, k)$

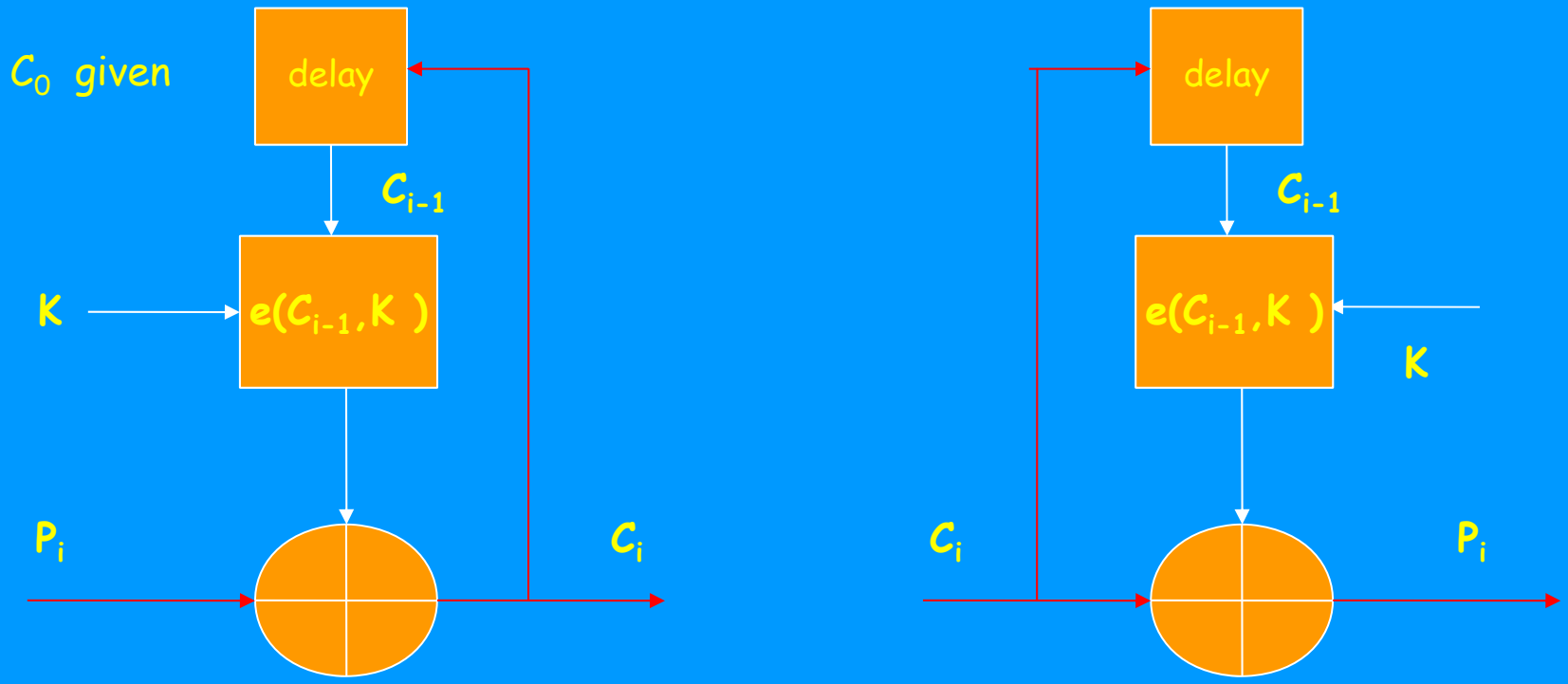
R_0 given

Advantage: no error propagation; easy to start decryption (using the encrpytion function) when time is known

Disadvantage: manipulation possible when parts of plaintext are known by using the XOR to replace P_i

Can you show this?

DES cipher feedback mode



Decypher mode →

$$C_i = P_i \oplus e(C_{i-1}, K)$$

$$P_i = C_i \oplus e(C_{i-1}, K)$$

DES cipher feedback mode

$C_i = P_i \oplus e(C_{i-1}, k)$

Receive: $C_i^* = C_i \oplus N_i$

$P_i^* = C_i^* \oplus e(C_{i-1}, k)$

$= C_i \oplus N_i \oplus e(C_{i-1}, k)$

$= P_i \oplus N_i$

$P_{i+1}^* = C_{i+1} \oplus e(C_i^*, k)$

$\neq P_{i+1}$

$P_{i+2} = C_{i+2} \oplus e(C_{i+1}, k)$ OK

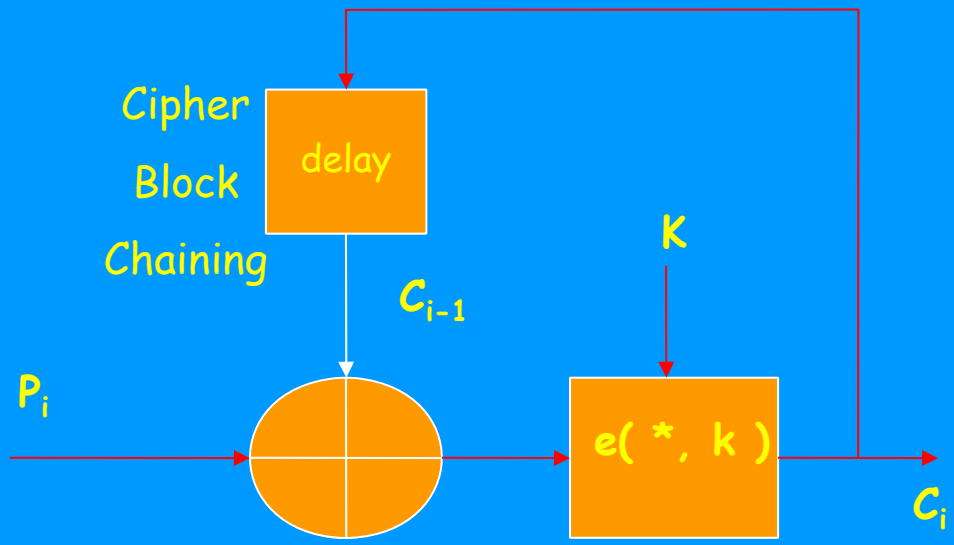
Advantage: Scalable feedback, easy to synchronize

disadvantage:

replacement of P_i gives unpredictable decryption of P_{i+1}

error propagation over two blocks

Cipher block chaining



C_0 given

$C_i = e(P_i \oplus C_{i-1}, k)$

$P_i = C_{i-1} \oplus d(C_i, k)$

Ciphertext depends on whole message

Advantage: identical blocks of plaintext will encrypt to different blocks of ciphertext.

The last cipher block can be used as a signature

Disadvantage: error propagation over two blocks

Cipher block chaining

$C_i = e(P_i \oplus C_{i-1}, k)$

$P_i = C_{i-1} \oplus d(C_i, k)$

Receive: $C_i^* = C_i \oplus N_i$

$P_i = C_{i-1} \oplus d(C_i^*, k)$

$P_{i+1} = C_i^* \oplus d(C_{i+1}, k)$

$P_{i+2} = C_{i+1} \oplus d(C_{i+2}, k)$

Advantage: identical blocks of plaintext will encrypt to different blocks of ciphertext.

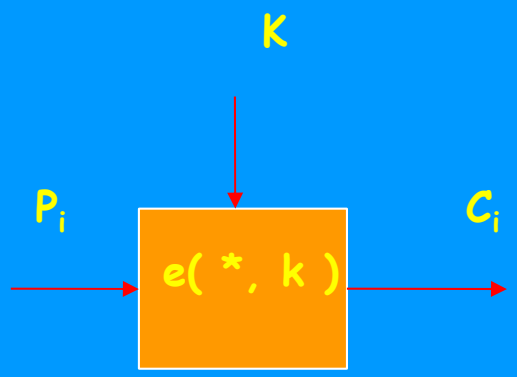
The last cipher block can be used as a signature

Disadvantage: error propagation over two blocks

Note: the errors in the second block are the same as the channel errors. The first block has about 50% errors., due to the decryption of an erroneous block

Cipher block chaining is much the most **widely used mode**.
- IPsec specifies it as the only permitted mode.
- PGP and TLS use it as well.

Electronic Codebook Mode

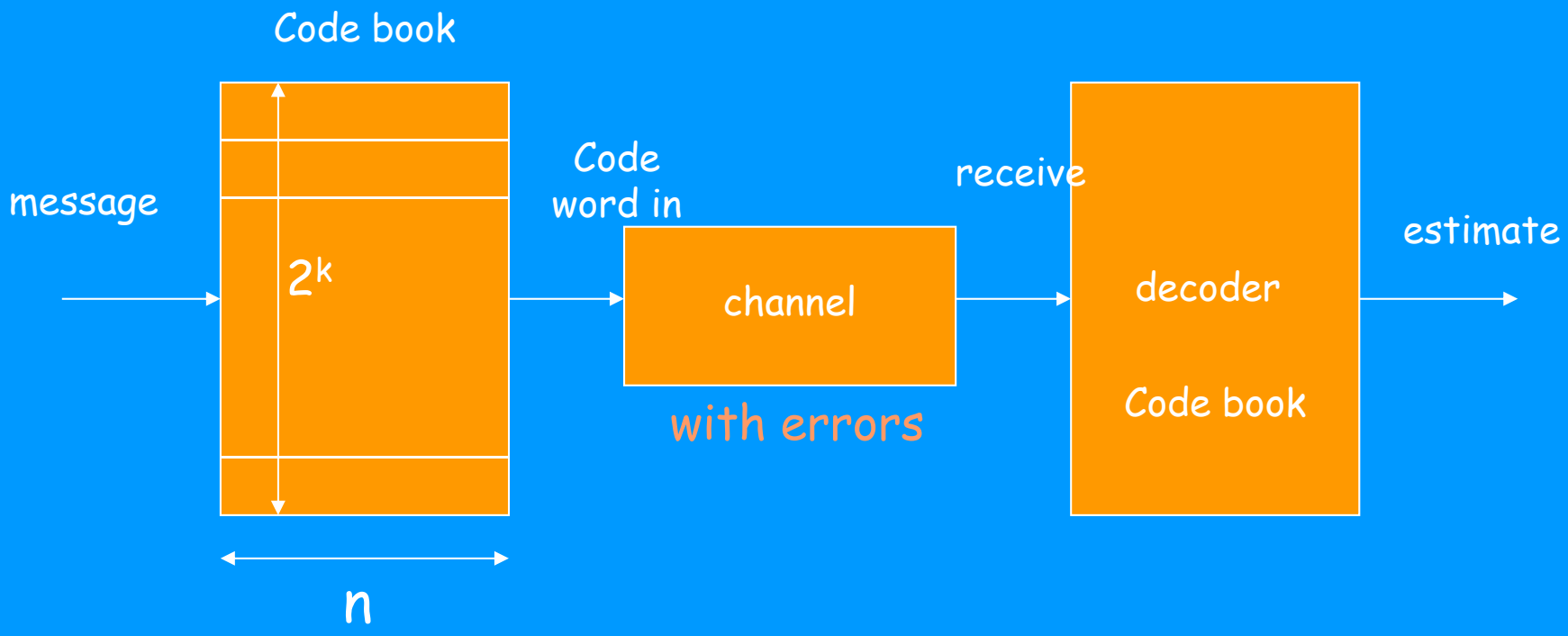


Advantage:
any block decrypted independently of the others.
lost data blocks do not affect the decryption of other blocks.

Disadvantage:
easy for known-plaintext attacks.
same block of plaintext results in same blocks of ciphertext
easy to substitute, rearange, delete or insert old blocks

Error correction

Practical communication system design



There are 2^k code words of length n

k is the number of information bits transmitted in n channel uses

error protection

- Obtained by Error Control Codes (ECC)
  - Forward Error Correction (FEC)
  - Error Detection and feedback (ARQ)

- Performance depends on error statistics!
  - Error models are very important

example

code words: 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 0

received: 0 0 0 1 1

difference: 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1

best guess: 0 1 0 1 1

only 1 difference

Definitions

- Hamming distance between $\underline{x}$ and $\underline{y}$ is

  $d_H := d(\underline{x}, \underline{y})$ is the # of positions where $x_i \neq y_i$

- The minimum distance of a code C is
  - $d_{min} = \min \{ d(\underline{x}, \underline{y}) \mid \underline{x} \in C, \underline{y} \in C, \underline{x} \neq \underline{y} \}$

- Hamming weight of a vector $\underline{x}$ is

  - $w(\underline{x}) := d(\underline{x}, \underline{0})$ is the # of positions where $x_i \neq 0$

Performance

A code with minimum distance d_{min} is capable of correcting t errors if

$$d_{min} \geq 2t + 1.$$

Proof: If $\leq$ t errors occur, then since $d_{min} \geq 2t + 1$ an incorrect code word has at least t+1 differences with the received word.

LINEAR CODE GENERATOR

The code words are

- linear combinations of the rows of a binary generator matrix *G* with dimensions k, n
- G must have rank k!

Example: Consider k = 3, n = 6.

generator matrix G =

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

(1,0,1)G = (0, 0, 1, 0, 1, 1)

Example (optimum)

- Single Parity check code $d_{min} = 2$, $k = n-1$

$G = [\, I_{n-1} \; P \,] =$

$$\left[\begin{array}{ccc|c} 100 & \bullet\bullet\bullet & 0 & 1 \\ 0100 & \bullet\bullet\bullet & 0 & 1 \\ \bullet\bullet\bullet & & & \\ & & & \\ 00 & \bullet\bullet\bullet & 01 & 1 \end{array}\right]$$

All codewords have even weight!

Example (optimum)

- Repetition code: $d_{min} = n$, $k = 1$

- $G = [\, 1 \;\; 1 \bullet\bullet\bullet 1 \,]$

Property

The **set of distances**

from all code words to the all zero code word

is the same as to any other code word.

Proof:

$d(\underline{x}, \underline{y}) = d(\underline{x} \oplus \underline{x}, \underline{z} = \underline{y} \oplus \underline{x}) = d(0, \underline{z})$,

by linearity $\underline{z}$ is also a code word.

Thus!

the **determination** of the minimum distance of a code is equivalent to

the **determination** of the minimum Hamming weight of the code words.

The complexity of this operation is proportional to # of code words

Some remarks

- Generators for different k and n
  - are constructed using mathematics
  - listed in many text books

What remains is the decoding!

Example k = 4, n = 7, m = 3

$$G = \begin{pmatrix} 1000 & 110 \\ 0100 & 101 \\ 0010 & 011 \\ 0001 & 111 \end{pmatrix}$$

$$k = 2^3 - 3 - 1 = 4$$

In Bluetooth we have a shortened (10,15) Hamming Code

Famous codes

- Famous codes are BCH code specified by the first row of the encoding matrix
  All following rows are cyclic shifts of this

Normally, BCH codes are specified for lengths $2^m - 1$

Example: length 127, $k = 120$, $d_{min} = 3$ correcting 1 error
$k = 113$, $d_{min} = 5$ correcting 2 errors

We can shorten the code to k = 64 and n = 71, to correct 1 error in a DES word

How do we shorten? Can you give a list of length 63 codes?

Error detection with the function f

We do all operations modulo 2 (XOR).

A packet has the form $C(X) = A(X)\ F(X)$

where

$F(X) = 1 + ... + X^{n-k}$ check polynomial

$A(X) = a_0 + a_1X + ... + a_{k-1}X^{k-1}$ information packet of length k.

$C(X) = c_0 + c_1X + ... + c_{n-1}X^{n-1}$ code word

detection procedure

Assume that we receive the polynomial $R(X) = C(X) + E(X)$

where $E(X)$ is a binary **error** polynomial

i.e. an error vector (0,0,0,1,0) $\rightarrow C(X) = X^3$

the decoder calculates R(X) mod F(X)

- if the result = 0 no error detected
- if the result ≠ 0, then an error is detected.

The polynomial R(X) mod F(X) = 0 iff E(X) is a multiple of F(X).

performance

ASSUME: the polynomial $F(X)$ has the form $F(X) = 1 + \ldots + X^{n-k}$

- degree n-k; - and a nonzero constant term,

THEN: any error burst of length $\leq$ n-k has a polynomial representation that looks like $E(X) = X^i (1 + \ldots + X^{n-k-1})$ and can thus never be a multiple of $F(X)$.

$F(X)$ is capable of detecting any burst of length $\leq$ n-k !

Some examples

- All coefficient operations are modulo-2
  - $(1+ X + X^3) (X + X^2)$ $= X+ X^2 + X^4 + X^2+ X^3 + X^5$
    $= X + X^3 + X^4 + X^5$
- errors change 0 ➡ 1 and 1 ➡ 0
- Ex: $(X + X^4 + X^5)$ modulo-$(1 + X^2) = 1$
  - Subtract as often as possible $(1 + X^2)$, but keep calculating mod-2

CRC standard polynomials

- CRC-8: $x^8+x^2+x^1+1$
- CRC-10: $x^{10}+x^9+x^5+x^4+x^1+1$
- CRC-12: $x^{12}+x^{11}+x^3+x^2+1$
- CRC-16: $x^{16}+x^{15}+x^2+1$
- CRC-CCITT: $x^{16}+x^{12}+x^5+1$
- CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+ x^{16}+x^{12}+x^{11}+x^{10}+x^8 x^7+x^5+x^4+x^2+1$

- Ethernet uses CRC-32
- HDLC: CRC-CCITT
- ATM: CRC-8, CRC-10, and CRC-32

Why these polynomials (CRC-16 and CRC-CCITT) ?

- 1st property: Multiples of these polynomials have even weight

  Hence: error vectors of odd weight cannot be a multiple!

- 2nd property: error vectors of the form $E(X) = (1 + X^i)$
  are not divisible by $F(X)$ for $i < L$,
  L larger than word length

  Try to check this property!

- **Conclusion:** 1, 2 and 3 errors are detectable. CHECK!