# Classical Cryptography

## Introduction: a puzzling matter?

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.5, 2020/02/05 15:55:14 UTC)

Monday, February 3, 2020

1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Outline


1. Organisation
    - Global Structure
    - Lectures
    - Practical exercises
    - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Outline


1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Organisation

- Information available on the OS3 Website/Wiki
  - `https://www.os3.nl/2019-2020/courses/crypto/start`
- Lectures
- Practical exercises
- Programming exercises

# Outline

1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Lectures

- Seven weeks
- February 3 - March 19
  - Monday, 15:00-17:00, G0.10-G0.12
  - Thursday, 15:00-17:00
    - February 6 and March 19: A1.28
    - February 13: A1.04
    - February 20, 27 and March 5: G0.23-0.25
- **No lecture** on March 12
- Q&A session on March 19

# Guest Lecture

- Monday, February 24
  - **Enigma**, by Hans van der Meer

# Outline

1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Practical and “programming” exercises

- Monday, 15:00-17:00, G0.10-G0.12
- Thursday, 15:00-17:00, G0.23-G0.25
- Lab assistant: **Felix Brakel**
- Programming language used is Ruby
  - You may replace it by something of your own choice
  - This is **not** a programming course
  - The programs are **tools** supporting cryptanalysis

# Outline


1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Judgment

- The final grading is only determined by the **written exam**
- Primary learning material
  - (Referenced parts of) Joshua Holden's **The Mathematics of Secrets**
  - **Slides** from the lectures
- Secondary learning material
  - Referenced parts of Hans van der Meer's **syllabus**
  - Material that can reasonably be expected to be known from practical and programming **exercises**

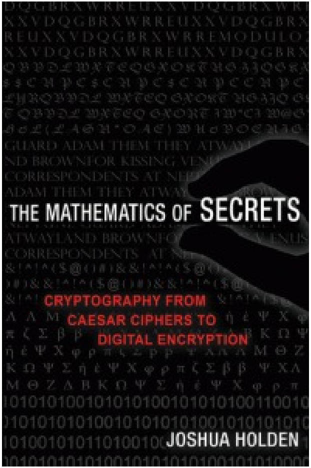
# Exam dates

- Classical Cryptography **exam** will be on
  - Monday, March 23, 09:00-12:00, Science Park C0.05
- Classical Cryptography **resit** will be on
  - Tuesday, May 26, 18:00-21:00, Science Park D1.111

# Outline

1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Book




- **The Mathematics of Secrets**: Cryptography from Caesar Ciphers to Digital Encryption
- Joshua Holden
- ISBN-13: 9780691141756 (hardcover)
- ISBN-13: 9780691183312 (paperback)
- `http://mathofsecrets.com/` (`https://mathofsecrets.com/?`)

# Outline


1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Some advice

- Keep up with theory and practice **right from the start**
- **Read** the book (like in a “flipped classroom”)

*The only true wisdom is in knowing you know nothing*

*—Socrates*

# Outline


1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# Basic terminology

cryptology cryptography plus cryptanalysis

cryptography secret writing

- steganography is hidden writing

cryptanalysis (unauthorized) reading of a cryptogram

- or even getting the key (possibly partially)
- or doing traffic analysis

# Basic symmetric/secret scheme

$$C = \mathcal{E}(M, K)$$

$$M = \mathcal{D}(C, K)$$

$$M = \mathcal{D}(\mathcal{E}(M, K), K)$$

- $\mathcal{E}$ is encryption; $\mathcal{D}$ is decryption
- $M$ is the message; $C$ is the cryptogram; $K$ is the key
- $\mathcal{E}(-, K)$ is injective for each K
- K has to be kept a secret between two communicating parties

# Basic asymmetric/public scheme

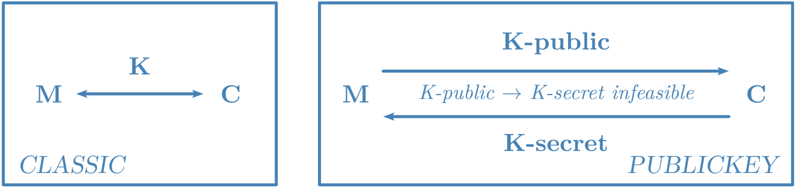
$$C = \mathcal{E}(M, K_p)$$

$$M = \mathcal{D}(C, K_s)$$

$$M = \mathcal{D}(\mathcal{E}(M, K_p), K_s)$$

- $\mathcal{E}$ is encryption; $\mathcal{D}$ is decryption
- $M$ is the message; $C$ is the cryptogram; $K$'s are two keys
- $\mathcal{E}(-, K_p)$ is injective for each $K_p$
- $K_s$ has to be kept a secret for each participant separately
- $K_p$ must be known to all parties (in a **verifiable** way)

# Symmetric versus asymmetric encryption



# Kerckhoffs' rules

- The system must be practically, if not mathematically, indecipherable.
- **It should not require secrecy, and it should not be a problem if it falls into enemy hands.**
- It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will.
- It must be applicable to telegraph communications.
- It must be portable, and should not require several persons to handle or operate.
- Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

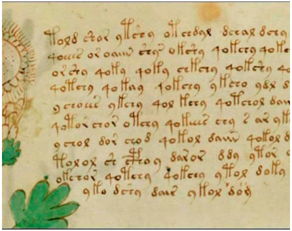
# Types of attack

- Increasing in strength:
  - Ciphertext-only
  - Known-plaintext
  - Chosen-plaintext
  - Chosen-ciphertext
- From observation to interaction
  - Passive (observing only)
  - Active (changing messages)

# Outline

1. Organisation
   - Global Structure
   - Lectures
   - Practical exercises
   - Examination
2. Book
3. Advice
4. Basic concepts in cryptography
5. Some examples
6. Puzzling

# The Voynich manuscript



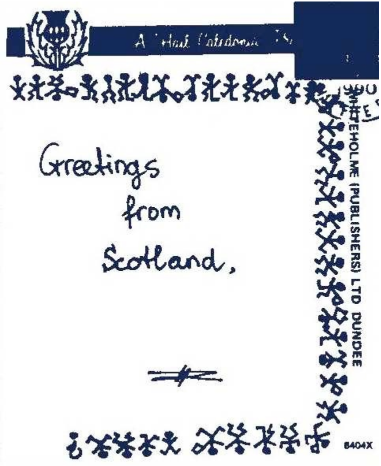
Real or fake?

Decoded or not?

Latest claims Nicholas Gibbs (September 2017)

and Greg Kondrak (January 2018, using AI)

# A personal message

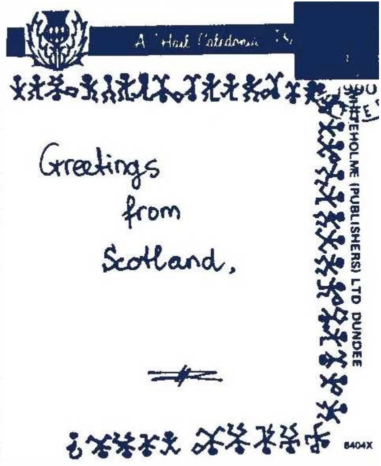



Source: Hans van der Meer

—

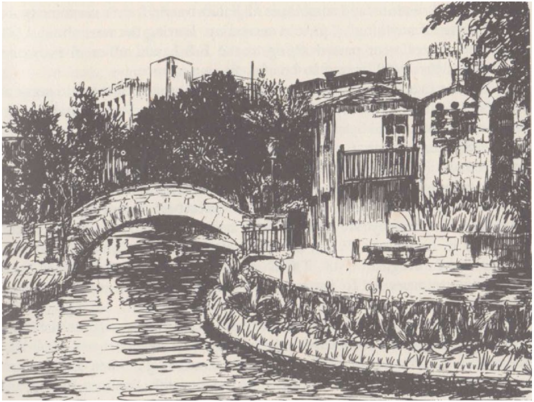
personal message

# A personal message




https://en.wikipedia.org/wiki/The_Adventure_of_the_Dancing_Men

# Just a picture?



Source: https://scienceblogs.de/klausis-krypto-kolumne/2015/05/21/versteckte-nachrichten-in-modezeichnungen-grashalmen-und-apfelbaeumen/

# Outline

1 Organisation
- Global Structure
- Lectures
- Practical exercises
- Examination

2 Book

3 Advice

4 Basic concepts in cryptography

5 Some examples

6 Puzzling

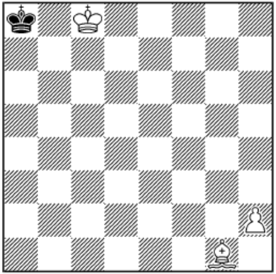
# Why puzzling?

- Accuracy
- Brain training
- Creativity
- Having fun
- Out of the box
- ...

These are all important for cryptanalysts

# Puzzle 1: Chess retrograde analysis

**Smullyan**

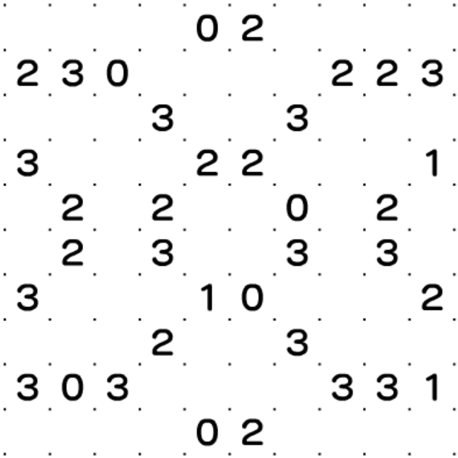


White to move. What was black's last move?

http://www.mathpuzzle.com/retrograde.html

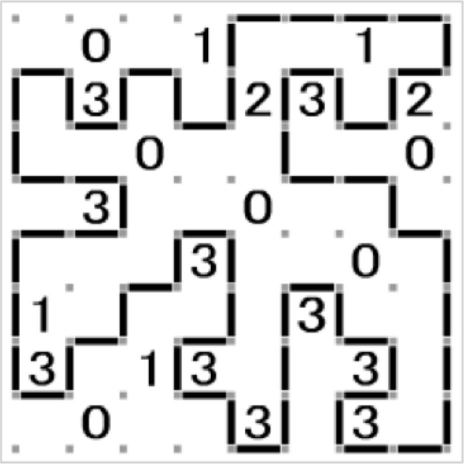
(https://www.mathpuzzle.com/retrograde.html?)

# Puzzle 2: Slitherlink




What are the rules of this game?

# Puzzle 2: Slitherlink continued




Try it yourself

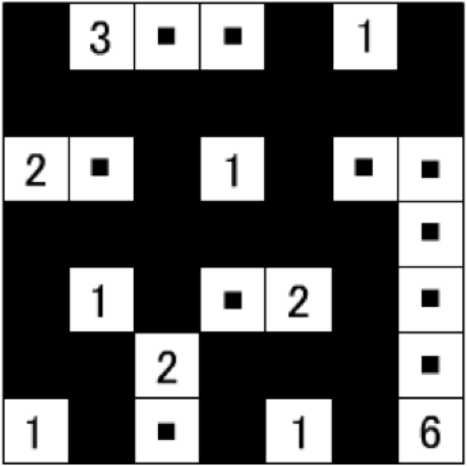
https://www.brainbashers.com/slitherlink.asp

# Puzzle 3: Nurikabe

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 5 | | 2 |
| 3 | | | | | | | | | |
| | 4 | | | 2 | | | | | |
| | | | | | | 3 | | | |
| | 4 | | | | 4 | | | | |
| | | | | | | | | | 3 |
| | | | | | | | | | |
| | | | | | | | | | |
| | 3 | | | 3 | | | | | |
| | | 1 | | | 1 | | 3 | | 3 |

What are the rules of this game?

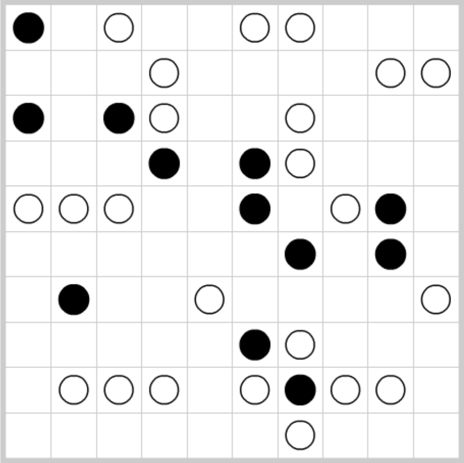
# Puzzle 3: Nurikabe continued




Try it yourself

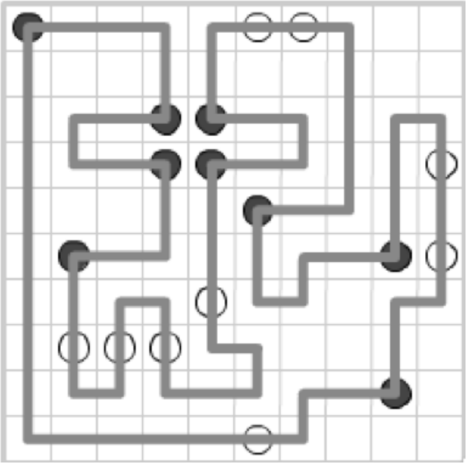
`https://www.brainbashers.com/nurikabe.asp`

# Puzzle 4: Masyu



What are the rules of this game?

# Puzzle 4: Masyu continued



Try it yourself

https://www.interactive.onlinemathlearning.com/fun_pearl.php

# Classical Cryptography

## Basics: monoalphabetic substitution

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.6, 2020/02/12 11:00:30 UTC)

Thursday, February 6, 2020

1 The classic Caesar substitution cipher
- Caesar's system
- Alphabet encoding
- Modular arithmetic
- Mathematical formulation
- Caesar cryptanalysis

2 General monoalphabetic systems
- Generating alphabets
- Some number theory
- Composition of ciphers

3 Extension of the alphabet
- Classic systems
- The Hill cipher

# Outline

1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

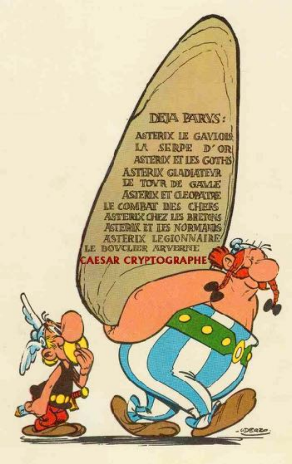
# Outline

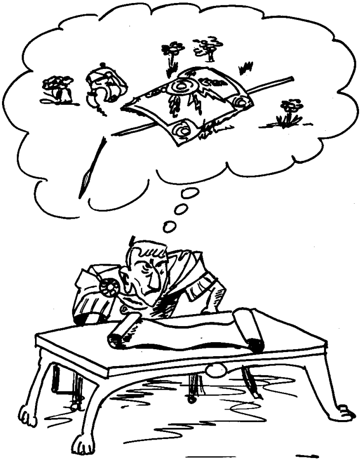
1 The classic Caesar substitution cipher
- Caesar's system
- Alphabet encoding
- Modular arithmetic
- Mathematical formulation
- Caesar cryptanalysis

2 General monoalphabetic systems
- Generating alphabets
- Some number theory
- Composition of ciphers

3 Extension of the alphabet
- Classic systems
- The Hill cipher

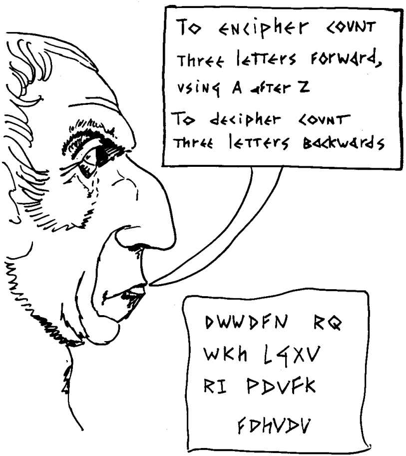
# Caesar wants to hide his plans



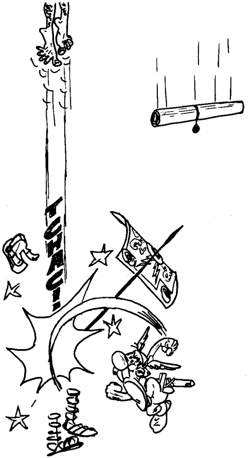
Source: Slides Hans van der Meer

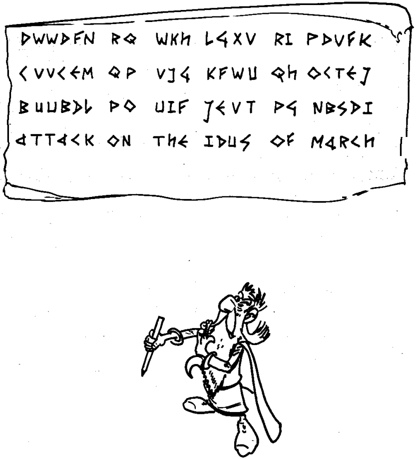
# Caesar's cryptosystem




Source: Slides Hans van der Meer

# Interception and cryptanalysis






Who notices the peculiarities here?

Source: Slides Hans van der Meer

# Caesar encryption

- Caesar encryption is a forward[1] rotation of the alphabet by 3 places

```
abcdefghijklmnopqrstuvwxyz

DEFGHIJKLMNOPQRSTUVWXYZABC
```

Figure 1: Rotation by 3 positions

- An example encryption

```
an example encryption

DQ HADPSOH HQFUBSWLRQ
```

Figure 2: Encryption of "an example encryption"

[1] Although historically, Suetonius mentions backward

# Caesar decryption

- Caesar decryption works by turning around the encryption process

| |
|---|
| DEFGHIJKLMNOPQRSTUVWXYZABC |
| abcdefghijklmnopqrstuvwxyz |

Figure 3: Encryption turned around (backward rotation by 3 places)

| |
|---|
| ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| xyzabcdefghijklmnopqrstuvw |

Figure 4: The same decryption reordered

# Outline

1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Encoding (numbering) the alphabet

| | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| modern | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| legacy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

- Modern mathematics starts counting at 0
- The legacy variant, starting at 1, is equivalent to ordering the alphabet as

  zabcdefghijklmnopqrstuvwxy

- This is because, when rotating the alphabet, we consider $26 = 0$

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Clock arithmetic

$24 = 0$ (or maybe $12 = 0$)

- $\mathbb{Z}_{24} = \{0, 1, 2, \ldots, 23\}$
- $23 + 1 \equiv 24 \equiv 0 \pmod{24}$

**Definition ($n \in \mathbb{N}, n > 1, a, b \in \mathbb{Z}$)**

$$a \equiv b \pmod{n} \iff n \mid (a - b) \iff \exists k \in \mathbb{Z}(k \cdot n = (a - b))$$

**Theorem**

*"$. \equiv . \pmod{n}$" is an* **equivalence** *relation on $\mathbb{Z}$ which is also a* **congruence**.

*$\mathbb{Z}_n$ is the set of integers modulo n.*

**Corollary**

*Addition and multiplication can be performed $\pmod{n}$ as usual.*

# Clock arithmetic

## Examples

$$22 + 5 \equiv 3 \pmod{24}$$

$$22 \cdot 5 \equiv 110 \equiv 14 \pmod{24}$$

$$-2 \cdot 5 \equiv -10 \equiv 14 \pmod{24}$$

$$2 \cdot 12 \equiv 24 \equiv 0 \pmod{24}$$

$$2 \not\equiv 0 \pmod{24}$$

$$12 \not\equiv 0 \pmod{24}$$

$\mathbb{Z}_{24}$ has "divisors of zero" or "zero divisors", which is considered an unwanted property in general.

# Clock arithmetic

## Convention

### $\pmod{n}$ as a function

The function application $a \pmod{n}$ means the unique $b$ such that $0 \leq b < n$ and $a \equiv b \pmod{n}$, as a relation.

- The use of $\pmod{n}$ both as a binary relation as well as a function can be confusing:

$$(a \pmod{n} \equiv a) \pmod{n}$$

$$a \pmod{n} = (a \pmod{n})$$

# Who's afraid of zero?

## or the AM/PM mess

- Splitting up 24 hours as $2 \cdot 12$ hours the sensible way
  - 0:00 AM (midnight), 1:00 AM, ..., 11:59 AM
  - 0:00 PM (midday, noon), 1:00 PM, ..., 11:59 PM
- Splitting up 24 hours as $2 \cdot 12$ hours the confusing way
  - 12:00 AM (midnight), 12:59 AM, 1:00 AM, ..., 11:59 AM
  - 12:00 PM (midday, noon), 12:59 PM, 1:00 PM, ..., 11:59 PM
  - $12 \equiv 0 \pmod{12}$, but $12 \not\equiv 0 \pmod{24}$,

    so using 12 hours in this context is confusing
    - It seems that in Japan 00:00 AM (12:00 PM) is midnight

      and 12:00 AM is noon

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Caesar mathematically

## Caesar encryption and decryption

$$\mathcal{E}(p) = (p + 3) \pmod{26} \tag{1}$$

$$\mathcal{D}(c) = (c - 3) \pmod{26} \tag{2}$$

- This works exactly the same with modern and legacy encoding
- Encryption and decryption is **keyless**
- Algorithm must be kept secret

# Caesar variants with a key

Let $k$ be a key, where $0 \leq k < 26$.

**Caesar encryption and decryption with key $k$**

$$\mathcal{E}_k(p) = (p + k) \pmod{26} \quad (3)$$

$$\mathcal{D}_k(c) = (c - k) \pmod{26} \quad (4)$$

- Even if the algorithm is known the key protects the encryption
- Since the key space is very small a brute force search is doable
- We call this is **shift cipher** or **additive cipher**

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Caesar brute force decrypting "VLONY ZILWY"

# Caesar brute force decrypting “VLONY ZILWY”

vlony zilwy

uknmx yhkvx

tjmlw xgjuw

silkv wfitv

rhkju vehsu

qgjit udgrt

pfihs tcfqs

oehgr sbepr

ndgfq radoq

mcfep qzcnp

lbedo pybmo

kadcn oxaln

jzcbm nwzkm

iybal mvyjl

hxazk luxik

gwzyj ktwhj

fvyxi jsvgi

euxwh irufh

dtwvg hqteg

csvuf gpsdf

brute force

aqtsd enqbd

zpsrc dmpac

yorqb clozb

xnqpa bknya

wmpoz ajmxz

# Caesar brute force decrypting “VLONY ZILWY”

vlony zilwy

uknmx yhkvx

tjmlw xgjuw

silkv wfitv

rhkju vehsu

qgjit udgrt

pfihs tcfqs

oehgr sbepr

ndgfq radoq

mcfep qzcnp

lbedo pybmo

kadcn oxaln

jzcbm nwzkm

iybal mvyjl

hxazk luxik

gwzyj ktwhj

fvyxi jsvgi

euxwh irufh

dtwvg hqteg

csvuf gpsdf

**brute force**

aqtsd enqbd

zpsrc dmpac

yorqb clozb

xnqpa bknya

wmpoz ajmxz

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Monoalphabetic substitution

## Definition

A monoalphabetic substitution is the systematic replacement of letters by other letters in a one-to-one way.

## Example monoalphabetic encryption and decryption

| | |
|---|---|
| abcdefghijklmnopqrstuvwxyz | ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| DJEHKVNIOLARUQXPYWGTCSMFZB | kzuacxsdhbejwgipnlvtmfroqy |

This example was generated using a Nomcom procedure with pool size 26 on input 1, 2, ..., 16[2]

[2] see RFC 3797

# Intermezzo: a real example (Spanish)

```
ADHRF SID QINVJX IH XDNAJIXJHAD
VFH YINEVJ YDZEVJHJ PFO J TTDPJX
J YE PDVEHJ JTTE DNAJ HFVWD DTTJ
DN YIO QFHEAJ O NEYLJAEVJ DNLDXF
WJVDXIHJ EYLXDNEFH QIDHJ
```

1. 1-letter word a or y sometimes o
2. 2-letter word u. usually un
3. 3-letter word ..e usually que
4. 4-letter word abbc usually alli or ella
5. Doubled start letter mostly l as in

   llegar, llevar, lleno, lluvia

# Generating a monoalphabetic substitution from a keyword

| |
|---|
| abcdefghijklmnopqrstuvwxyz |
| KEYWORDABCFGHIJLMNPQSTUVXZ |

Figure 5: Using "KEYWORD" as the keyword

| |
|---|
| abcdefghijklmnopqrstuvwxyz |
| REPATDLSBCFGHIJKMNOQUVWXYZ |

Figure 6: Using "REPEATED LETTERS" as the keyword/keyphrase

# Generating a monoalphabetic substitution using decimation

| abcdefghijklmnopqrstuvwxyz |
|---|
| EJOTYDINSXCHMRWBGLQVAFKPUZ |

Figure 7: Encoding using a **multiplicative cipher** (legacy)

| abcdefghijklmnopqrstuvwxyz |
|---|
| AFKPUZEJOTYDINSXCHMRWBGLQV |

Figure 8: Encoding using a **multiplicative cipher** (modern)

- A multiplicative cipher is also called a **decimation**

# Decoding of these multiplicative ciphers

| |
|---|
| ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| upkfavqlgbwrmhcxsnidytojez |

Figure 9: Decoding of the **multiplicative cipher** (legacy)

| |
|---|
| ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| avqlgbwrmhcxsnidytojezupkf |

Figure 10: Decoding of the **multiplicative cipher** (modern)

- The encoding factor was 5. What is the decoding factor?

# Mathematical description of decimation

## Multiplicative encryption and decryption

$$\mathcal{E}_e(p) = ep \pmod{26} \tag{5}$$

$$\mathcal{D}_d(c) = dc \pmod{26} \tag{6}$$

- There is now a difference between modern and legacy encoding
- Modern encoding works best for programming
- $d$ is the **multiplicative inverse**[3] of $e$

[3] Does this always exist?

# Outline


1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Greatest common divisor

## An example of Euclid's algorithm

We want to find the gcd (greatest common divisor) of 49 and 35:

### Euclid's reduction

$$49 = 1 \cdot 35 + 14 \implies \text{gcd}(49, 35) = \text{gcd}(35, 14)$$

$$35 = 2 \cdot 14 + 7 \implies \text{gcd}(35, 14) = \text{gcd}(14, 7)$$

$$14 = 2 \cdot 7 + 0 \implies \text{gcd}(14, 7) = \text{gcd}(7, 0) = 7$$

### Euclid's reversal

$$7 = 35 - 2 \cdot 14 \quad \wedge \quad 14 = 49 - 1 \cdot 35$$

$$\begin{aligned} 7 &= 35 - 2 \cdot (49 - 1 \cdot 35) \\ &= -2 \cdot 49 + 3 \cdot 35 \end{aligned}$$

# Greatest common divisor

## Euclid's algorithm

### Theorem

*For all $a, b \in \mathbb{Z}$ we can (effectively) find $p, q \in \mathbb{Z}$ such that*

$$gcd(a, b) = p \cdot a + q \cdot b$$

*Finding $p$ and $q$ can be done using Euclid's algorithm and reversal.*

### Definition

$a$ and $b$ are called **relatively prime** iff $\gcd(a, b) = 1$.

### Theorem

*If $a$ and $b$ are relatively prime (the extended) Euclid's algorithm calculates $p$ and $q$ such that*

$$p \cdot a + q \cdot b = 1$$

# Application to decimation

In our example we had $e = 5$ and we want to find its inverse $d$ modulo 26.

## Calculation of inverse of 5 modulo 26

$$26 = 5 \cdot 5 + 1 \implies 1 = 26 + (-5) \cdot 5$$

So the inverse of 5 modulo 26 is -5 (or 21).

- A decimation's inverse is another decimation,

  just with a different multiplication factor.
- What happens if $e$ and 26 are not relatively prime?
- This explains why the decoding described earlier is indeed

  just a decimation with factor 21

# Outline

1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Combining multiple ciphers

- Combining two shift ciphers with key $k_1$ and $k_2$
  - Result is shift cipher with key $k_1 + k_2$
- Combining two decimations with key $e_1$ and $e_2$
  - Result is decimation with key $e_1 e_2$
- Combining a decimation with key $e$ and a shift with key $k$
  - First decimate, then shift gives the **affine cipher**

    defined by $\mathcal{E}_{e,k}(p) = ep + k \pmod{26}$
  - First shift, then decimate gives the cipher

    defined by $\mathcal{E}_{e,k}(p) = e(p + k) \pmod{26}$

    or $\mathcal{E}_{e,k}(p) = ep + ek \pmod{26}$, just another affine cipher

# Outline

1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis
2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers
3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

# Extending the "alphabet"

- Until now substitutions are **monographic**
  - One letter of the alphabet is replaced with another letter
- What happens if we "extend the alphabet" (make it **polygraphic**)?
  - For instance replace a combination of two letters of the alphabet by another combination of two letters (so using **digraphs**)
  - Effectively this extends our alphabet from 26 to $26 \cdot 26 = 676$ "letters" (or symbols, atoms, literals, ...)
  - The number of possible (monoalphabetic) substitions increases from $26! = 403291461126605635584000000$ to $676! \approx 1.8837 \cdot 10^{1621}$

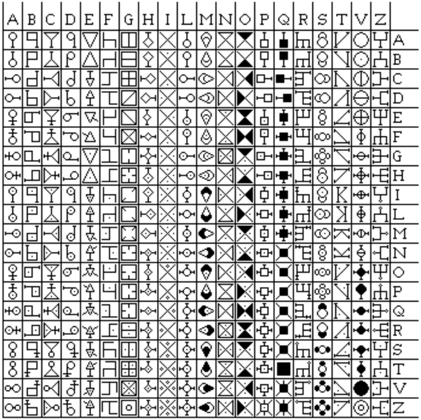
# Outline

1. The classic Caesar substitution cipher
   - Caesar's system
   - Alphabet encoding
   - Modular arithmetic
   - Mathematical formulation
   - Caesar cryptanalysis

2. General monoalphabetic systems
   - Generating alphabets
   - Some number theory
   - Composition of ciphers

3. Extension of the alphabet
   - Classic systems
   - The Hill cipher

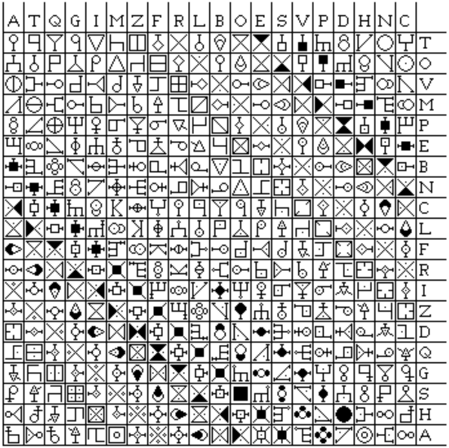
# Giovanni Batista della Porta's digraph encoding




Source: `http://www.quadibloc.com/crypto/pp010302.htm`

(Can you spot anomalies?)

# Giovanni Batista della Porta’s digraph encoding




Source: Slides Hans van der Meer

# An example digraph substitution

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | LZ | SW | BH | YJ | YR | WP | BC | FB | FW | XH | DY | MV | KC | UL | CJ | FJ | XW | BR | AD | JP | BJ | PM | JW | IU | OU | DE |
| B | AJ | GE | KT | AP | TN | VO | GY | CT | JS | OB | YM | MH | WJ | PF | PA | TA | IF | NR | GG | PV | LH | NX | KX | ST | UT | RP |
| C | LW | KW | DO | QF | JN | LX | DI | TL | DR | RM | SS | HF | RB | QU | UJ | KR | MY | GO | WF | TG | RS | YQ | SC | FI | CR | HK |
| D | UV | FP | PS | XZ | EV | GR | SV | KF | ZX | WL | RU | WO | YZ | JJ | NJ | VJ | IT | QT | XG | AS | KE | WE | ND | HS | YC | UH |
| E | AO | CZ | CI | SI | BV | OM | ZO | LE | GD | LB | OI | UK | RC | DK | PZ | YX | KJ | ZT | EM | BS | IZ | XL | RF | WA | YW | EL |
| F | OC | RI | SP | FY | VH | QE | SE | FC | IK | NZ | RG | LN | TX | NM | SD | JB | UQ | XY | ZG | ML | AV | JC | QM | PQ | AB | ZF |
| G | MD | VE | FX | MW | OD | PJ | XX | HT | IC | LC | NH | ZD | GC | YY | VP | YA | PC | BE | JF | DS | QK | SX | EQ | ET | YD | JH |
| H | BT | TK | PR | KY | EC | AN | HZ | SO | YV | MF | ES | YP | FU | AK | NI | SJ | YT | LY | TF | KV | NV | XV | DJ | WX | OO | QB |
| I | WR | GK | IE | QH | EZ | OY | MU | MT | LA | BP | HA | NM | TJ | QJ | AL | EE | SU | GA | HI | MG | YO | GW | KS | AY | JE | NO |
| J | VG | ZY | UE | FM | EH | FR | ZW | CA | DN | WD | KD | AU | GP | YS | XM | MR | NC | BQ | HC | NS | NN | ZJ | GJ | VB | RA | TH |
| K | KQ | UR | VQ | AT | OA | YI | FS | RJ | LT | JD | KI | PG | AC | MI | CD | BG | TZ | PH | OT | WQ | IH | LK | OK | XE | HY | CX |
| L | YE | VX | GS | VY | IM | HW | HB | JX | NE | ZI | IB | HL | BI | QO | VK | AH | LL | VT | YB | DL | ZC | QI | JA | DH | UY | ZH |
| M | HU | EW | UC | IJ | UO | SQ | OR | EP | ZE | MX | KL | IQ | TS | QZ | BM | TI | JV | VD | XS | OH | IX | TV | TB | QN | UW | KN |
| N | LM | CB | SK | EY | PO | FQ | LG | MS | RK | VS | RW | CL | II | RO | ZR | NP | HX | RN | BF | IV | DX | XI | UG | BX | JM | AQ |
| O | TQ | XN | SH | ZS | WK | OX | WU | HH | MQ | PT | GL | QA | EX | PX | ZB | HJ | VW | SB | PL | DB | NA | CM | UX | IA | JK | LU |
| P | XD | GM | TC | FG | EJ | FN | WT | NF | OG | QY | DZ | NB | NU | IN | ZV | HM | CS | JU | WV | QG | FH | RQ | TE | DA | GH | AF |
| Q | YG | DV | EF | HV | TU | HR | LJ | CQ | FK | VC | GF | FZ | ER | XK | NW | XU | VA | ED | MN | UI | RL | GX | WH | WS | TM | OW |
| R | OS | XR | ID | SG | CY | TY | KG | ZN | YL | KZ | OJ | GU | VF | VR | BD | JO | GV | ZU | FF | WG | XF | GZ | KP | KU | QD | JT |
| S | RY | GQ | ZZ | HP | CC | HQ | UF | AD | PK | DW | XQ | DU | RH | DC | GN | QR | DM | MK | SF | RZ | MC | FT | BZ | LQ | IO | LO |
| T | YF | BA | UU | YN | TR | LD | WB | NQ | TW | VN | RD | FA | YU | OP | OQ | LR | FL | JI | JZ | HO | QQ | QC | GI | QW | KH | MA |
| U | JQ | XO | CH | EA | SY | XJ | IG | PD | ZL | LF | LP | KO | JY | ZP | UD | KA | TD | NG | ZQ | CF | AI | XT | HD | XB | UB | CW |
| V | XC | EI | BU | VV | AX | DF | MZ | VU | VM | RV | UP | PN | WC | FE | DT | IL | ZM | CU | EK | WZ | OF | LS | BL | IS | XA | BB |
| W | LI | FO | KM | JR | CV | QP | EG | WN | UA | NT | AG | UN | KK | US | WY | MP | SL | MB | BK | KB | AR | YH | DD | OE | DG | VI |
| X | AE | FD | ZK | SA | QX | SM | HE | CE | ZA | QV | IY | CN | PY | HN | JG | XP | AZ | UZ | BN | BW | PI | MO | AW | QL | DP | HG |
| Y | RX | NY | TO | MJ | SR | PE | BO | TT | BY | OV | WM | VZ | GT | CO | JL | GB | SN | NK | OL | PU | EU | RE | PP | RT | AM | CG |
| Z | ON | ME | IP | PB | WI | EB | LV | PW | EN | VL | NL | AA | QS | WW | RR | SZ | DQ | UM | CP | TP | IW | YK | CK | OZ | FV | IR |

Source: Slides Hans van der Meer

(Can you spot anomalies?)

# Playfair square with keyword

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

Figure 11: Playfair square (keyword STRANDBAL) (Charles Wheatstone, 1854)

Source: Slides Hans van der Meer

# Playfair (row based) substitutions

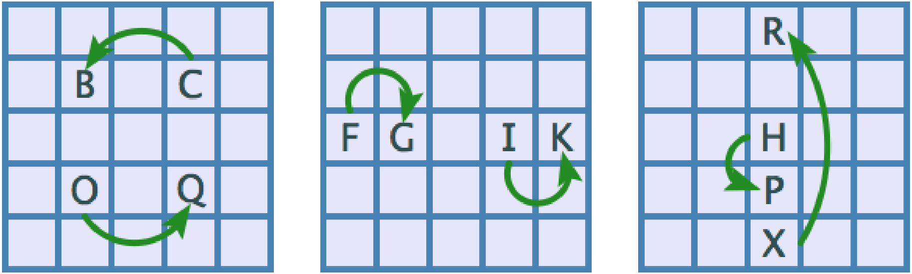



Figure 12: Playfair encryption (OC→QB; FI→GK; HX→PR)

Source: Slides Hans van der Meer

# Outline

1 The classic Caesar substitution cipher
- Caesar's system
- Alphabet encoding
- Modular arithmetic
- Mathematical formulation
- Caesar cryptanalysis

2 General monoalphabetic systems
- Generating alphabets
- Some number theory
- Composition of ciphers

3 Extension of the alphabet
- Classic systems
- The Hill cipher

# The (affine) Hill cipher

- Based on linear algebra
- Considers polygraphs as vectors
- An affine cipher built from
  - An (invertible) matrix
  - A translation vector
  - All modulo the size of the base alphabet

$$\begin{pmatrix} 3 & 5 \\ 6 & 1 \end{pmatrix} \begin{pmatrix} 10 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 10 \end{pmatrix} \pmod{26}$$

# Decoding the Hill cipher uses inverse matrix

- Encoding

$$\mathcal{E}(p_1, p_2) = \begin{pmatrix} 3 & 5 \\ 6 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} \pmod{26}$$

- Decoding

$$\mathcal{D}(c_1, c_2) = \begin{pmatrix} -1 & 5 \\ 6 & -3 \end{pmatrix} \left[ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right] \pmod{26}$$

# Classical Cryptography

## Monoalphabetic cryptanalysis

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.2, 2020/02/08 13:18:13 UTC)

Monday, February 10, 2020

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests

2. Example

3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# Outline

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

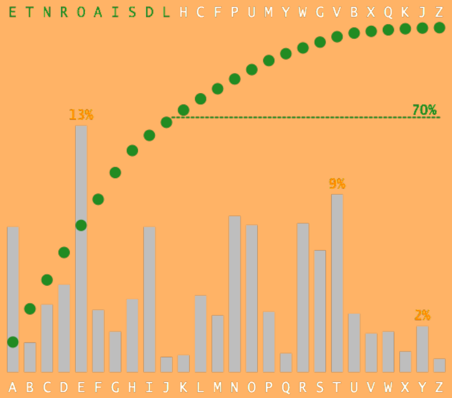
# Outline

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# Letter frequencies

- A simple method to attack monoalphabetic ciphers
  - **letter frequency analysis**
- Some letters occur more (or less) than others
  - This is (somewhat) language dependent

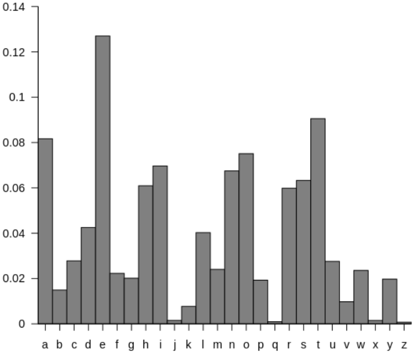
# Letter frequency diagram



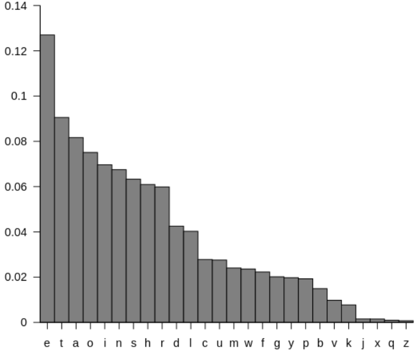

Source: Slides Hans van der Meer

Unknown language or text source

# English letter frequency




Ordered by alphabet




Ordered by frequency

Source: `<https://en.wikipedia.org/wiki/Letter_frequency>`

# Outline

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# The index of coincidence (IoC)

- Introduced by **William Friedman**
- Probability that two letters chosen randomly from a text, based on an alphabet of $n$ letters, are the same
- Given probabilities $p_0, \dots, p_{n-1}$ for the $n$ letters
  - $\text{IoC} = \sum_{i=0}^{n-1} p_i^2$
- For text with a (uniformly) random frequency distribution this reduces theoretically (obviously) to $1/n$ ($\approx 0.038$ for $n = 26$)
- For an English text (with the English frequency distribution) this amounts to $\approx 0.066$, found by doing experiments

# The $\phi$-test

- The IoC clearly distinguishes English text from random text
- Friedman observed that the IoC is

  **invariant under monoalphabetic substitution**

- Using the IoC to check for monoalphabeticity is called the $\phi$-test
- For an unknown ciphertext of length $M$ this test calculates
  - $\text{IoC} = \sum_{t=A}^{Z} f_t(f_t - 1)/M(M-1)$
  - Here $f_t$ is the number of occurrences of the letter $t$
  - For small texts the $-1$ is used to avoid counting identity as equality
    - hence letters that occur only once don't contribute to the IoC

# Breaking Caesar (by hand and automatically)

- Brute force 26 keys and see if you get plaintext (we did this before)
- Match (visually) the frequency distribution of the cryptogram to standard English by shifting the frequency graph
- To automate this the $\phi$-test doesn't help, use the $\chi$-test instead
  - The $\chi$-test is also called cross-product sum
  - Consider two texts f and g of length M and N, respectively and calculate $\sum_{t=A}^{Z} f_t g_t / MN$
  - Find highest $\chi$ value for comparison between shifted frequency diagram of cryptogram and English text

# Breaking monoalphabetic substitutions

- First use the $\phi$-test to check for monoalphabeticity
- Order the ciphertext letter distribution by frequency

  and try to match this with standard English

  (or whatever language you may suspect is being used)
- Look at digraph (or even trigraph) frequencies
- Look at beginning and ending of words (different frequencies)
- Check vowels versus consonants and other letter patterns
- Look at keywords for alphabet construction
- Try to find cribs

# Outline

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

```
QBVDL WXTEQ GXOKT NGZJQ GKXST RQLYR
XJYGJ NALRX OTQLS LRKJQ FJYGJ NGXLK
QLYUZ GJSXQ GXSLQ XNQXL VXKOJ DVJNN
BTKJZ BKPXU LYUNZ XLQXU JYQGX NTYQG
XKXQJ KXULK QJNQN LQBYL OLKKX SJYQG
XNGLU XRSBN XOFUL YDSXU GJNSX DNVTY
RGXUG JNLEE SXLYU ESLYY XUQGX NSLTD
GQXKB AVBKX JYYBR XYQNQ GXKXZ LNYBS
LRPBA VLQXK JLSOB FNGLE EXYXU LSBYD
XWXKF SJQQS XZGJS XQGXF RLVXQ BMXXK
OTQKX VLJYX UQBZG JQXZL NG
```

# Exercise 1

## Exercise 1

- Count letters and make a table of frequencies
- Generate a frequency diagram, using a spreadsheet
- Calculate the Index of Coincidence
- Is it an additive cipher?
- Try to solve the cryptogram by assuming it is affine

# Outline


1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# Outline

1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# Homophones

- Homophones
  - A classic way to flatten frequency distributions
  - Introduce more than one ciphertext letter option for some of the plaintext letters
    - Especially for plaintext letters with high frequency
    - Needs a larger ciphertext alphabet
  - This is an example where the encryption function may be randomized (to a small extent)

# Math of Secrets: 2.2 homophones

```
IW*CI W@G*L &H&L( ASN*A E)U&V $CNPC
SIW*E DDSA@ LTCIH !(A#C V%EIW *!#HA
*IW@N TAEHR $CI(C JTS!C SHDS# SIW@S
DVW@R G$HH* SIW*W )JH@( CUGDC IDUIW
*&AIP GWTUA TLS$L CIW*D IWTG! #HATW
TRG$H H*SQT U$G*I W@S)D GHWTR APBDG
*S%EI W@WDB @HIG@ IRWWX H&CV+ XHWVG
*LLXI WW#HE G)VG@ HHI#A AEGTH @CIAN
W*L!H Q%I!L )DAAN R)BTI B)K#C VXC#I
HDGQX ILXIW IW@VA *&B!C SIWTH E**S$
UA(VW I
```

# Exercise 2

## Exercise 2

- Count symbols and make a table of frequencies
- Generate a frequency diagram, using a spreadsheet
- Calculate the Index of Coincidence for all symbols
- Calculate the Index of Coincidence for only the letters
- Is it a monoalphabetic cipher?
- Identify homophones and solve the cryptogram

# Outline


1. Statistical Cryptanalysis
   - Frequencies
   - The index of coincidence: $\phi$- and $\chi$-tests
2. Example
3. Countermeasures against statistical cryptanalysis
   - Homophones
   - Polyalphabetic substitutions

# Polyalphabetic substitutions

**Definition**

A **polyalphabetic substitution** is the replacement of letters by other letters by using a (possibly) different alphabet for each plaintext letter

- Poly**alphabetic** uses different alphabets per plaintext letter
- Poly**graphic** uses a larger alphabet for plaintext and ciphertext
- Poly**literal** uses a larger alphabet for ciphertext only

# Classical Cryptography

## Polyalphabetic substitution

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.1, 2020/02/08 13:40:46 UTC)

Tuesday, February 10, 2020

1. Early polyalphabetic systems
2. Later polyalphabetic systems
3. Variations
4. A few related systems

# Outline


1. Early polyalphabetic systems
2. Later polyalphabetic systems
3. Variations
4. A few related systems

# Polyalphabetic ciphers

- Use more than one (cipher) alphabet
- Use a changing cipher alphabet (often for each plaintext letter)
- Leon Battista **Alberti** (1404 – 1472)
  - Cipher disk
- Johannes **Trithemius** (1462 – 1516)
  - Tabula recta
- Giovan Battista **Bellaso** (1505 – ca 1575)
  - Keyed polyalphabetic cipher
- Giambattista della **Porta** (ca 1535 – 1615)
  - Porta reduced table

# Leon Battista Alberti (1404 – 1472)


- *De Cifris (On Ciphers)*
- Cipher disk
- Regularly change cipher alphabet
- Communicate change in ciphertext
- Outer ring plaintext inner ring ciphertext

# Johannes Trithemius (1462 – 1516)

- Tabula recta
  - "proper table"
  - square table
  - letter square
  - tableau
- Progressive system
  - The cipher alphabet changes each letter by taking the next line in the tabula recta

# Tabula recta

```
      A B C D E F G H I K L M N O P Q R S T U X Y Z W
    +================================================
 0  | A B C D E F G H I K L M N O P Q R S T U X Y Z W
 1  | B C D E F G H I K L M N O P Q R S T U X Y Z W A
 2  | C D E F G H I K L M N O P Q R S T U X Y Z W A B
 3  | D E F G H I K L M N O P Q R S T U X Y Z W A B C
 4  | E F G H I K L M N O P Q R S T U X Y Z W A B C D
 5  | F G H I K L M N O P Q R S T U X Y Z W A B C D E
 6  | G H I K L M N O P Q R S T U X Y Z W A B C D E F
 7  | H I K L M N O P Q R S T U X Y Z W A B C D E F G
 8  | I K L M N O P Q R S T U X Y Z W A B C D E F G H
 9  | K L M N O P Q R S T U X Y Z W A B C D E F G H I
10  | L M N O P Q R S T U X Y Z W A B C D E F G H I K
11  | M N O P Q R S T U X Y Z W A B C D E F G H I K L
12  | N O P Q R S T U X Y Z W A B C D E F G H I K L M
13  | O P Q R S T U X Y Z W A B C D E F G H I K L M N
14  | P Q R S T U X Y Z W A B C D E F G H I K L M N O
15  | Q R S T U X Y Z W A B C D E F G H I K L M N O P
16  | R S T U X Y Z W A B C D E F G H I K L M N O P Q
17  | S T U X Y Z W A B C D E F G H I K L M N O P Q R
18  | T U X Y Z W A B C D E F G H I K L M N O P Q R S
19  | U X Y Z W A B C D E F G H I K L M N O P Q R S T
20  | X Y Z W A B C D E F G H I K L M N O P Q R S T U
21  | Y Z W A B C D E F G H I K L M N O P Q R S T U X
22  | Z W A B C D E F G H I K L M N O P Q R S T U X Y
23  | W A B C D E F G H I K L M N O P Q R S T U X Y Z
```

Figure 1: Original tabula recta (no J, V; W at end)

```
         A B C D E F G H I J K L M N O P Q R S T U V W X Y Z --> plaintext alphabet
       +=====================================================
     0 | A B C D E F G H I J K L M N O P Q R S T U V W X Y Z \
     1 | B C D E F G H I J K L M N O P Q R S T U V W X Y Z A  |
     2 | C D E F G H I J K L M N O P Q R S T U V W X Y Z A B  |
     3 | D E F G H I J K L M N O P Q R S T U V W X Y Z A B C  |
     4 | E F G H I J K L M N O P Q R S T U V W X Y Z A B C D  |
     5 | F G H I J K L M N O P Q R S T U V W X Y Z A B C D E  |
     6 | G H I J K L M N O P Q R S T U V W X Y Z A B C D E F  |
 p   7 | H I J K L M N O P Q R S T U V W X Y Z A B C D E F G  |
 r   8 | I J K L M N O P Q R S T U V W X Y Z A B C D E F G H  |
 o   9 | J K L M N O P Q R S T U V W X Y Z A B C D E F G H I  |
 g  10 | K L M N O P Q R S T U V W X Y Z A B C D E F G H I J  |
 r  11 | L M N O P Q R S T U V W X Y Z A B C D E F G H I J K  |
 e  12 | M N O P Q R S T U V W X Y Z A B C D E F G H I J K L  |
 s  13 | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M   \ ciphertext
 s  14 | O P Q R S T U V W X Y Z A B C D E F G H I J K L M N   / alphabets
 i  15 | P Q R S T U V W X Y Z A B C D E F G H I J K L M N O  |
 o  16 | Q R S T U V W X Y Z A B C D E F G H I J K L M N O P  |
 n  17 | R S T U V W X Y Z A B C D E F G H I J K L M N O P Q  |
 s  18 | S T U V W X Y Z A B C D E F G H I J K L M N O P Q R  |
    19 | T U V W X Y Z A B C D E F G H I J K L M N O P Q R S  |
    20 | U V W X Y Z A B C D E F G H I J K L M N O P Q R S T  |
    21 | V W X Y Z A B C D E F G H I J K L M N O P Q R S T U  |
    22 | W X Y Z A B C D E F G H I J K L M N O P Q R S T U V  |
    23 | X Y Z A B C D E F G H I J K L M N O P Q R S T U V W  |
    24 | Y Z A B C D E F G H I J K L M N O P Q R S T U V W X  |
    25 | Z A B C D E F G H I J K L M N O P Q R S T U V W X Y /
```

Figure 2: Modern (progressive) tabula recta

# Periodic progressive systems

- Normal progression $0, 1, 2, \ldots$ is very regular
  - Its period is 26
- To make things less predictable you can vary the progression
  - A step pattern like $1, 3, 2$ generates the irregular progression $0, 1, 4, 6, 7, 10, 12, \ldots$
  - The progression index (PGI) is $1 + 3 + 2 = 6$
  - Now the period turns out to be 39

# Kryha encryption device

- Mechanical device making irregular steps when pushing a lever
  - With 17-steps pattern $7, 6, 7, 5, 6, 7, 6, 8, 6, 10, 5, 6, 5, 7, 6, 5, 9$
  - The period is an impressive $17 \cdot 26 = 442$

# Kryha cryptanalysis

- Cryptanalysis by William Friedman and his team
  - William Friedman, Solomon Kullback,
    Frank Rowlett and Abraham Sinkov
- The challenge given was a 1135 letter cryptogram
- The challenge was broken (without computers)
  in a mere 2 hours and 41 minutes

# Giovan Battista Bellaso (1505 – ca 1575)

- "Forgotten by history"
- Introduced the keyed polyalphabet
  - Repeating-key cipher
  - Later named after Blaise de **Vigenère**
- Used reciprocal alphabets
  - Makes encryption and decryption identical operations
  - Later named after Francis **Beaufort**

# Outline

1 Early polyalphabetic systems

2 Later polyalphabetic systems

3 Variations

4 A few related systems

# Blaise de Vigenère (1523 – 1596)

- Used Bellaso's ideas
- Combined the following ideas
  - Tabula recta (now called Vigenère square)
  - Repeating-key cipher
- Plaintext letters along the top of the diagram
- Ciphertext letters inside the table
- Key letters along the left side of the diagram
  - Key letter equals first letter of cipher alphabet

```
                     plaintext

       A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
     +====================================================
   A | A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
   B | B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
   C | C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
   D | D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
   E | E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
   F | F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
   G | G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
   H | H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
k  I | I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
e  J | J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
y  K | K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
   L | L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
l  M | M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
e  N | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
t  O | O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
t  P | P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
e  Q | Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
r  R | R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
   S | S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
   T | T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
   U | U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
   V | V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
   W | W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
   X | X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
   Y | Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
   Z | Z A B C D E F G H I J K L M N O P Q R S T U V W X Y
```

Figure 3: Vigenère table (modern encoding)

# Mathematical formulation of Vigenère's encryption

- Let $P = P_0 P_1 \ldots P_{n-1}$ be the plaintext
- Let $K = K_0 K_1 \ldots K_{p-1}$ be the key with **period p**
- Then the cryptogram $C = C_0 C_1 \ldots C_{n-1}$ is given by
  - $C_i = \mathcal{E}_i(P_i) = P_i + K_{i \pmod p} \pmod{26}$
- For decryption we conclude
  - $P_i = \mathcal{D}_i(C_i) = C_i - K_{i \pmod p} \pmod{26}$
- Exchanging encryption and decryption is called “Variant Vigenère”

# More room for confusion

- We want to keep the simple mathematical relationship between plaintext letter and cryptogram letter: $C = P + K$
- And we also want to use legacy encoding
- The only way this works is by using an **alternative** Vigenère
- This non-standard table is what is used in the book
- In this case the key letters are not the first elements of the cipher alphabet

```
                          plaintext

         A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
       +=====================================================
     A | B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
     B | C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
     C | D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
     D | E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
     E | F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
     F | G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
     G | H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
     H | I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
k    I | J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
e    J | K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
y    K | L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
     L | M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
l    M | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
e    N | O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
t    O | P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
t    P | Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
e    Q | R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
r    R | S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
     S | T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
     T | U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
     U | V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
     V | W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
     W | X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
     X | Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
     Y | Z A B C D E F G H I J K L M N O P Q R S T U V W X Y
     Z | A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Figure 4: Alternative Vigenère table (legacy encoding; used in book)

# Francis Beaufort (1774 – 1857)

- Changes Vigenère square by starting with a mixed cipher alphabet
  - Which is a Caesar (key = 1) shift of the atbash cipher
  - Or if you want the atbash of a Caesar (key = -1) shift
- In modern encoding the Beaufort starting cipher alphabet
  - can also be described simply as a multiplicative cipher with factor -1
- In legacy encoding the Beaufort starting cipher alphabet
  - must be described by a more complicated affine cipher with
    - factor -1
    - additive 2

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

Figure 5: Beaufort table

# Mathematical formulation of Beaufort's encryption

- Let $P = P_0 P_1 \dots P_{n-1}$ be the plaintext (in modern encoding)
- Let $K = K_0 K_1 \dots K_{p-1}$ be the key with **period p**
- Then the cryptogram $C = C_0 C_1 \dots C_{n-1}$ is given by
  - $C_i = \mathcal{E}_i(P_i) = -P_i + K_{i \pmod{p}} \pmod{26}$
- For decryption we conclude
  - $P_i = \mathcal{D}_i(C_i) = -C_i + K_{i \pmod{p}} \pmod{26}$
- Now we clearly see the symmetric role of encryption and decryption
  - $P_i + C_i = C_i + P_i = K_{i \pmod{p}} \pmod{26}$

# Outline


1. Early polyalphabetic systems
2. Later polyalphabetic systems
3. Variations
4. A few related systems

# Giambattista della Porta (ca 1535 – 1615)

- Introduced the first digraph substitution
  - *De furtivis Literarum Notis* (1563)
  - His scientific work on cryptography
- Introduced another polyalphabetic cipher based on a reduced size table
  - Porta's reduced table

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

Figure 6: Full Porta table

# Reduced Porta table

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AB | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| CD | O | P | Q | R | S | T | U | V | W | X | Y | Z | N |
| EF | P | Q | R | S | T | U | V | W | X | Y | Z | N | O |
| GH | Q | R | S | T | U | V | W | X | Y | Z | N | O | P |
| IJ | R | S | T | U | V | W | X | Y | Z | N | O | P | Q |
| KL | S | T | U | V | W | X | Y | Z | N | O | P | Q | R |
| MN | T | U | V | W | X | Y | Z | N | O | P | Q | R | S |
| OP | U | V | W | X | Y | Z | N | O | P | Q | R | S | T |
| QR | V | W | X | Y | Z | N | O | P | Q | R | S | T | U |
| ST | W | X | Y | Z | N | O | P | Q | R | S | T | U | V |
| UV | X | Y | Z | N | O | P | Q | R | S | T | U | V | W |
| WX | Y | Z | N | O | P | Q | R | S | T | U | V | W | X |
| YZ | Z | N | O | P | Q | R | S | T | U | V | W | X | Y |

Figure 7: Reduced Porta table

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

Figure 8: “Plain mixed up”-table

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

Figure 9: “Cipher mixed up”-table

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

Figure 10: “Cipher and key mixed up”-table

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

Figure 11: “Cipher and key mixed up (sorted)”-table

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

Figure 12: “Plain, cipher and key mixed up”-table

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

Figure 13: “Plain, cipher and key mixed up (sorted)”-table

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

Figure 14: "Plain, cipher and key mixed up (all sorted)"-table

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

Figure 15: “Same mixed (Beaufort-style)” table

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

Figure 16: “Same mixed (Beaufort style and sorted)”-table

# Outline


1. Early polyalphabetic systems
2. Later polyalphabetic systems
3. Variations
4. A few related systems

# Multiplex systems (1): Alphabet strips

- Choose a set of alphabet strips from a given collection
- Each strip contains (two copies of) a permutation of the alphabet
- Put the plaintext inside one of the columns
- Read off the ciphertext from any other column
  - Each of the other 25 columns is called a **generatrix**

# Alphabet strip example

```
               plain           crypto
               v               w
11 A L T M S X V Q P N O H U W D I Z Y C G K R F B E J
 3 C Z I N X F Y Q R T V W L A D K O M J U B G E P H S
23 J C P G B Z A X K W R E V D T U F O Y H M L S I Q N
12 V E W O A M N F L H Q G C U J T B Y P Z K X I S R D
 7 V R O G S Y D U L C F M Q T W A H X J E Z B N I K P
       w       v
11 M S X V Q P N O H U W D I Z Y C G K R F B E J A L T
 3 Q R T V W L A D K O M J U B G E P H S C Z I N X F Y
23 X K W R E V D T U F O Y H M L S I Q N J C P G B Z A
12 X I S R D V E W O A M N F L H Q G C U J T B Y P Z K
 7 B N I K P V R O G S Y D U L C F M Q T W A H X J E Z
               v       w
11 F B E J A L T M S X V Q P N O H U W D I Z Y C G K R
 3 F Y Q R T V W L A D K O M J U B G E P H S C Z I N X
23 J C P G B Z A X K W R E V D T U F O Y H M L S I Q N
12 H Q G C U J T B Y P Z K X I S R D V E W O A M N F L
 7 T W A H X J E Z B N I K P V R O G S Y D U L C F M Q
               v                   w
11 I Z Y C G K R F B E J A L T M S X V Q P N O H U W D
```

Figure 17: Encryption of VYAND NADERT WATER into DDTJW XTWSI VKRZI X

Source: Syllabus Hans van der Meer

# Alphabet strips (M-138-A)

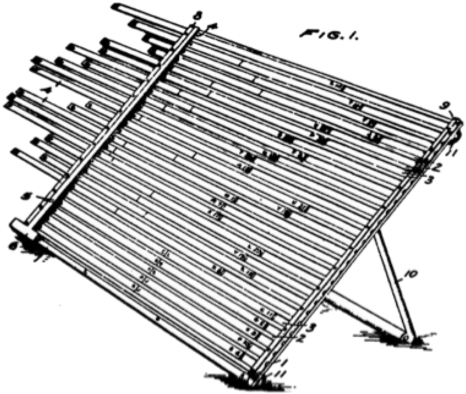



Figure 18: William Friedman's alphabet strips device

Source: Syllabus Hans van der Meer

# Multiplex systems (2): Jefferson cylinder (M-94)

- This is based on the same idea as the alphabet strips
- The alphabets are circumscribed on wheels mounted on a cylinder

Source: Syllabus Hans van der Meer

# Rotor based systems

- Similar to a progressive system based on a mixed cipher alphabet
- The difference is that it also has a “regressive” component
- In fact the next cipher alphabet is a **conjugation**

  of the previous cipher alphabet with a “Caesar-1” cipher
- Let R be the (arbitrary) rotor permutation and

  C an additive permutation with addition 1
- Then after k rotation steps the permutation is given by

$$R_k = C^{-k} \circ R \circ C^k$$

# Classical Cryptography

## Polyalphabetic cryptanalysis

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.3, 2020/02/12 10:36:42 UTC)

Thursday, February 13, 2020

1. Effect on the Index of Coincidence

2. Determination of the period

3. Composition of polyalphabetic ciphers

# Outline


1. Effect on the Index of Coincidence
2. Determination of the period
3. Composition of polyalphabetic ciphers

# The IoC of a polyalphabetic cipher (1)

- Assume the text length is n and the period is p
  - For simplicity suppose p divides n
- Let $\kappa_r$ be the IoC of random text ($\approx$0.038)
- Let $\kappa_e$ be the IoC of English plaintext ($\approx$0.066)
- If we split up the cryptogram in p columns
  - then each column of size n/p is monoalphabetic in itself
  - and letters in different columns seem unrelated

# The IoC of a polyalphabetic cipher (2)

So if we pick two different letters from the cryptogram

we expect an index of coincidence of (approximately)

$$\text{IoC} \approx \frac{n(n-\frac{n}{p})\kappa_r + n(\frac{n}{p}-1)\kappa_e}{n(n-1)}$$

or

$$\text{IoC} \approx \frac{n-\frac{n}{p}}{n-1}\kappa_r + \frac{\frac{n}{p}-1}{n-1}\kappa_e$$

- For $p = n$ this reduces to $\kappa_r$ (random)
- For $p = 1$ this reduces to $\kappa_e$ (monoalphabetic)

# Outline

1. Effect on the Index of Coincidence
2. Determination of the period
3. Composition of polyalphabetic ciphers

# Determination of an unknown period (1)

Solving for p and writing $\kappa_i$ for the IoC

we get from the previous estimation

$$p \approx \frac{\kappa_e - \kappa_r}{\kappa_i - \kappa_r + \frac{\kappa_e - \kappa_i}{n}}$$

So if n is large enough this reduces to

$$p \approx \frac{\kappa_e - \kappa_r}{\kappa_i - \kappa_r} \approx \frac{0.028}{\kappa_i - 0.038}$$

# Determination of an unknown period (2)

- The **Kasiski test**
- Look for repetitions of groups of letters in the cryptogram
- And how far they are apart: call this d for distance
- Probably the repetitions come from a repetition in the plaintext
- In that case d is a multiple of the period p
- A probable p follows from the consideration of all those d's
- **Charles Babbage** (1791 – 1871) probably invented this method years before **Friedrich Kasiski** (1805–1881) did

# The Kasiski method

Adapted from slides by Hans van der Meer

# Kasiski method

Until 1863 Vigenère is "le chiffre indéchiffrable"

Then major Friedrich Kasiski publishes

*"Die Geheimschriften und die Dechiffrier-kunst"*

a method to determine the period

uses repetitions in phase with this period

William F. Friedman, Riverbank Publication nr 22, 1920

*The Index of Coincidence and its Application in Cryptography*

# Repetitions

```
pt: EENCURSUSVANHETMATHEMATISCHCENTRUM
 k: STOEIPOESSTOEIPOESSTOEIPOESSTOEIPO
ct: WXBGCGGYKNTBLMIAELZXAEBXGGZUXBXZJA
```

real

```
pt: EENCURSUSVANHETMATHEMATISCHCENTRUM
 k: STOEIPOESSTOEIPOESSTOEIPOESSTOEIPO
ct: WXBGCGGYKNTBLMIAELZXAEBXGGZUXBXZJA
```

fake

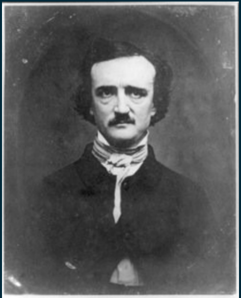
```
pt: EENCURSUSVANHETMATHEMATISCHCENTRUM
 k: STOEIPOESSTOEIPOESSTOEIPOESSTOEIPO
ct: WXBGCGGYKNTBLMIAELZXAEBXGGZUXBXZJA
```

coinci
dental

# Kulp message

```
Ge Jeasgdxv,

Zij gl mw, laam, xzy zmlwhfzek
ejlvdxw kwke tx lbr atgh lbmx
aanu bai Vsmukkss pwn vlwk agh
gnumk wdlnzweg jnbxvv oaeg enwb
zwmgy mo mlw wnbx mw al pnfdcfpkh
wzkex hssf xkiyahul. Mk num yexdm
wbxy sbc hv wyx Phwkgnamcuk?
```

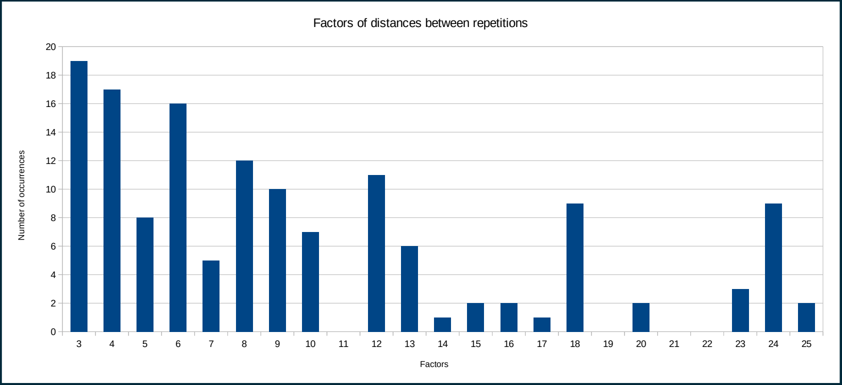


1839 from Kulp, Lewiston, Pennsylvania, USA
to Edgar Allen Poe, ed. Alexander’s Weekly Messenger

Note: jeasgdxv should be ieiasgdxv

# Kasiski analysis

zij gl mw, laam, xzy zmlwhfzek ejlvdxw kwke tx lbr atgh lbmx aanu bai vsmukkss pwn vlwk agh gnumk wdlnzweg jnbxvv oaeg enwb zwmgy mo mlw wnbx mw al pnfdcfpkh wzkex hssf xkiyahul mk num yexdm wbxy sbc hv wyx phwkgnamcuk



# 3 letters = THE ?

zij gl mw, laam, xzy zmlwhfzek ejlvdxw kwke tx lbr atgh lbmx aanu bai vsmukkss pwn vlwk agh gnumk wdlnzweg jnbxvv oaeg enwb zwmgy mo mlw wnbx mw al pnfdcfpkh wzkex hssf xkiyahul mk num yexdm wbxy sbc hv wyx phwkgnamcuk

XYZ = the → key letters

# Position on period 12

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **I** | **J** | | | | | | | | | | **Z** | ZIJ |
| **Y** | | | | | | | | | | **X** | **Z** | XZY |
| **B** | **R** | | | | | | | | | | **L** | LBR |
| | | **B** | **A** | **I** | | | | | | | | BAI |
| | **P** | **W** | **N** | | | | | | | | | PWN |
| | | | | | | | | **A** | **G** | **H** | | AGH |
| | | | | | | | | | **M** | **L** | **W** | MLW |
| **N** | **U** | **M** | | | | | | | | | | NUM |
| **S** | **B** | **C** | | | | | | | | | | SBC |
| | | | | | **W** | **Y** | **X** | | | | | WYX |

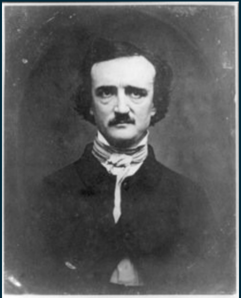
# Key letters

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **B** | **F** | | | | | | | | | | **G** | ZIJ |
| **U** | | | | | | | | | | **E** | **S** | XZY |
| **U** | **N** | | | | | | | | | | **S** | LBR |
| | | **I** | **T** | **E** | | | | | | | | BAI |
| | **W** | **P** | **J** | | | | | | | | | PWN |
| | | | | | | | | **H** | **Z** | **D** | | AGH |
| | | | | | | | | | **T** | **E** | **S** | MLW |
| **U** | **N** | **I** | | | | | | | | | | NUM |
| **Z** | **U** | **Y** | | | | | | | | | | SBC |
| | | | | | **D** | **R** | **T** | | | | | WYX |
| **U** | **N** | **I** | **T** | **E** | **D** | **R** | **T** | **A** | **T** | **E** | **S** | |

Note: The R in DRT should be DST and is one of the many mistakes in the cryptogram

# Kulp message decoded

Mr Alexander,
how ys it, that, the messenger
arrives here at the sace time
with the Saturgay cou rier and
other satuzdao paters when
avco rdidg to the cate it is
publishrd three days previous. Is
the fault witg you or tge
Possmastyrs?



Note the many mistakes (introduced by the editor?)

# Determination of an unknown period (3)

- The $\kappa$ **test**
- Friedman's original application of the theory of coincidence
- This time we look at **two** texts
  - that we compare **character by character**
- We expect coincidences $\kappa_r$ and $\kappa_e$ for respectively two random and two English texts
- The trick is to compare some cryptogram with a **displaced** (shifted, slid) copy of **itself**
- If the displacement is a multiple of the period coincidences rise

# Superimposition

- Knowing the period we can **superimpose** (Dutch: "in diepte leggen") the cryptogram
- Each column is monoalphabetic
- This makes cryptanalysis easy if the cipher is based for instance on a Vigenère with plain alphabet
- Each monoalphabet is then additive and we need only one letter for each column to determine it
- Simple letter frequency counts usually suffice

# Outline

1. Effect on the Index of Coincidence
2. Determination of the period
3. Composition of polyalphabetic ciphers

# Repeating-key framework for compositions

- Repeating-key polyalphabetic ciphers
- Each monoalphabetic cipher is either
  - Additive
    - So this is a standard Vigenère
  - Affine
    - The first cipher alphabet is mixed up by a decimation

# Keywords of the same length

- Composition gives a similar cipher
- The combined keyword length stays the same
    - Composition of additives stays additive
        - The keyword is the addition of keywords
        - Which makes it somewhat harder-to-guess
    - Composition of affines stays affine
        - The keyword is a linear combination of keywords
        - Also the decimation changes
        - Can you find out the exact formulas?

# Keywords of different lengths

- Let the length of the keywords K and L be $a$ and $b$ respectively
- Let lcm($a$,$b$) be the least common multiple of $a$ and $b$
- Let $a' = \text{lcm}(a, b)/b$ and $b' = \text{lcm}(a, b)/a$
- Reduce this situation to keywords of the same length
  - Consider keywords KK...K ($b'$ times) and LL...L ($a'$ times)
  - This results in two keywords of equal length lcm($a$,$b$)

# Classical Cryptography

## Basics: transpositions

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.3, 2020/02/12 10:37:03 UTC)

Thursday, February 13, 2020

1. Theoretic considerations
    - Permutations
2. Framework for cryptography
3. Transpositions
    - Geometric/Route ciphers
    - Permutation ciphers
    - Historic examples

# Outline

1. Theoretic considerations
   - Permutations

2. Framework for cryptography

3. Transpositions
   - Geometric/Route ciphers
   - Permutation ciphers
   - Historic examples

# Transpositions versus Substitutions

- Substitutions
  - transform objects into or replace objects by other objects
  - keeping the positions of these objects the same
- Transpositions
  - put the objects into a different position
  - keeping the identity of these objects the same
- Both operations can be represented by permutations
  - with paying careful attention to the semantics of each permutation **and its inverse**

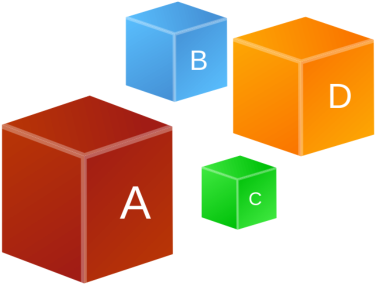
# Outline

1. Theoretic considerations
   - Permutations

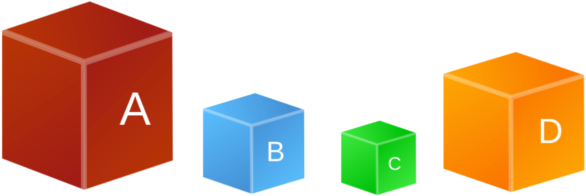
2. Framework for cryptography

3. Transpositions
   - Geometric/Route ciphers
   - Permutation ciphers
   - Historic examples

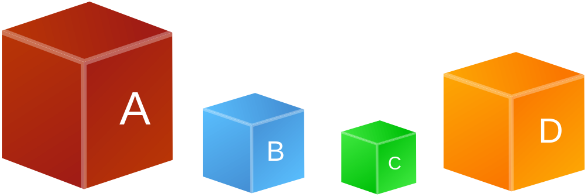
# Is this a permutation?



# And now?

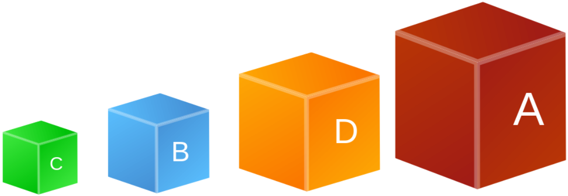


# And now?

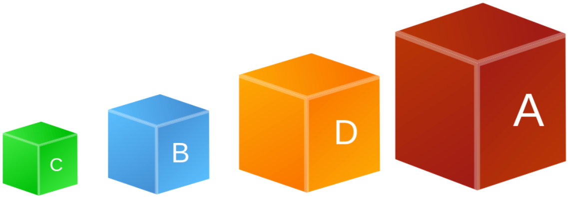



Do we order by identity, alphabetically? A<B<C<D?

# Or maybe like this?

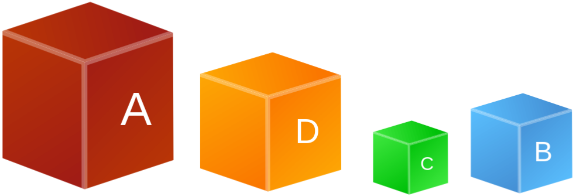


# Or maybe like this?

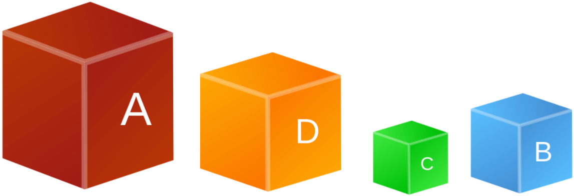



Do we order by size? tiny<small<big<huge (C<B<D<A)?

# And more options...

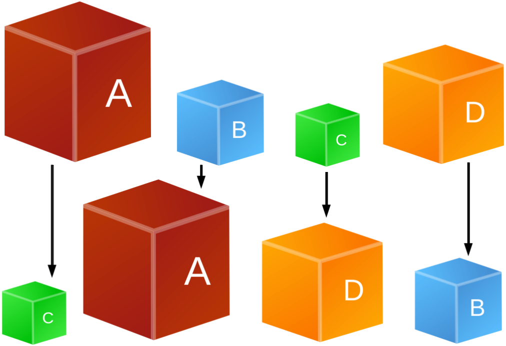


# And more options...




Ordered by color: red<orange<green<blue (A<D<C<B)

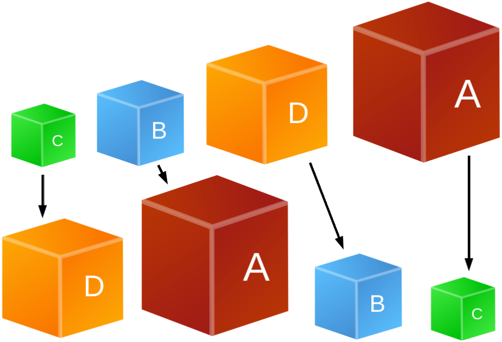
# Keeping position, but changing identity (1)




$$\begin{pmatrix} A & B & C & D \\ C & A & D & B \end{pmatrix} \qquad \begin{pmatrix} A & B & C & D \\ C & A & D & B \end{pmatrix}$$

1 2 3 4 1 2 3 4

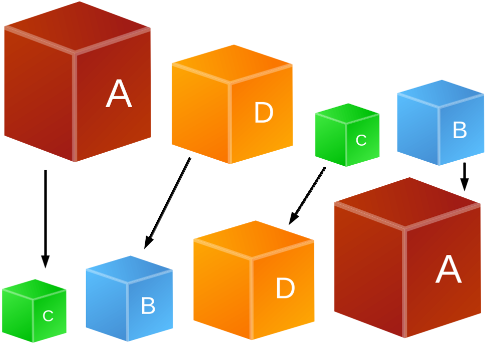
# Keeping position, but changing identity (2)




$$\begin{pmatrix} C & B & D & A \\ D & A & B & C \end{pmatrix} \qquad \begin{pmatrix} A & B & C & D \\ C & A & D & B \end{pmatrix}$$

1 2 3 4      4 2 1 3

# Keeping position, but changing identity (3)




$$\begin{pmatrix} A & D & C & B \\ C & B & D & A \end{pmatrix} \qquad \begin{pmatrix} A & B & C & D \\ C & A & D & B \end{pmatrix}$$

$$1 \quad 2 \quad 3 \quad 4 \qquad\qquad 1 \quad 4 \quad 3 \quad 2$$

# Legacy and modern notation for positions

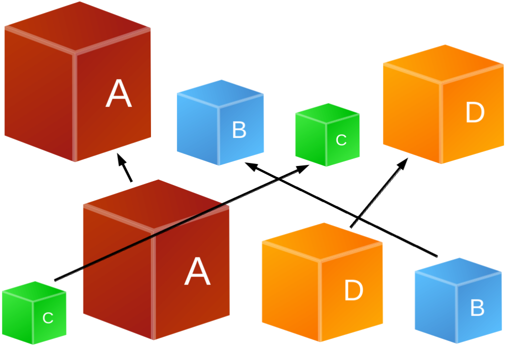
- It would be consistent to also use modern notation for positions and their permutations, like this

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 3 & 2 & 0 \end{pmatrix}$$

- But this time I give in to the notation used in the book in order not to create more confusion at this stage, hence

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}$$

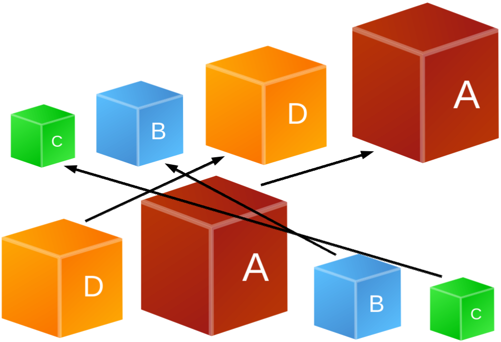
# Keeping identity, but changing position (1)




$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

$C \quad A \quad D \quad B$

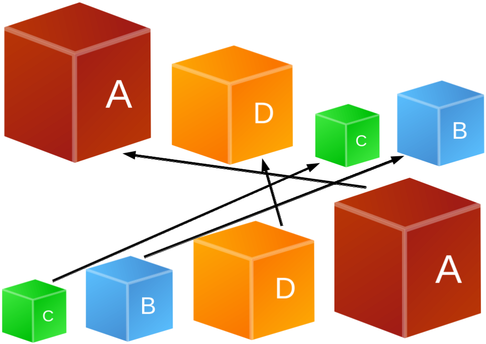
# Keeping identity, but changing position (2)




$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

*D* *A* *B* *C*

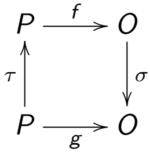
# Keeping identity, but changing position (3)




$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

$C \quad B \quad D \quad A$

# Commuting diagram (before: f; after: g)




- P is the set of positions
- O is the set of objects (or identities)
- $\sigma$ is a substitution (of identity) permutation
- $\tau$ is a transposition (change of position) permutation

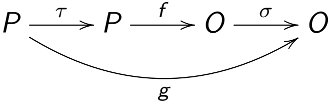
# Permutation directions

- Consider the transformation from top (f) to bottom (g)
- $g = \sigma \circ f \circ \tau$
- $g^{-1} = \tau^{-1} \circ f^{-1} \circ \sigma^{-1}$
- Notice the direction of the permutations and their inverses
  - The substitution permutation from top (f) to bottom (g)
  - The transposition permutation from bottom (g) to top (f)
- When reversing top and bottom things turn around
  - $f = \sigma^{-1} \circ g \circ \tau^{-1}$
  - $f^{-1} = \tau \circ g^{-1} \circ \sigma$

# Outline

1. Theoretic considerations
   - Permutations
2. Framework for cryptography
3. Transpositions
   - Geometric/Route ciphers
   - Permutation ciphers
   - Historic examples

# Texts as strings or sequences of characters




- P is n, where $n = \{0, \ldots, n-1\}$
- O is {A, B, C, ..., X, Y, Z}, our alphabet
- f is the plaintext[1] before encryption
    - or the ciphertext before decryption
- $g = \sigma \circ f \circ \tau$ is the ciphertext after encryption
    - or the plaintext after decryption

[1] Note that in this case f doesn't need to be injective nor surjective

# Relation between transposition and substitution

- Even though transposition and substitution seem unrelated they are kind of dual to each other
- Both are permutations (of position, respectively identity)
- Transposition contributes to Shannon's "diffusion"
- Substitution contributes to Shannon's "confusion"
- Taken together we might[2] call them
  - **su(b)positions**
  - **trans(s)titutions**

[2] This is in no way standard or accepted terminology

# Relaxing bijectivity of substitutions and transpositions

$$P' \underset{\tau}{\overset{\tau'}{\leftrightarrows}} P \xrightarrow{f} O \underset{\sigma}{\overset{\sigma'}{\leftrightarrows}} O'$$

- In the case that f is the given plaintext
  - $\sigma$ only needs to be injective, but not surjective
    - $\mathrm{id}_O = \sigma' \circ \sigma$
    - $O'$ can be “bigger” than $O$
  - $\tau$ only needs to be surjective, but not injective
    - $\mathrm{id}_P = \tau \circ \tau'$
    - $P'$ can be “bigger” than $P$

# Expansion and compression

- Suppose that $P'$ is indeed "bigger" than $P$
- We might use the following terminology if $\mathrm{id}_P = \tau \circ \tau'$
  - $\tau$ is an **expansion** (function)
  - $\tau'$ is the corresponding **compression** (function)
  - Mathematically speaking $\tau'$ is a **section** of $\tau$

# Outline


1. Theoretic considerations
   - Permutations
2. Framework for cryptography
3. Transpositions
   - Geometric/Route ciphers
   - Permutation ciphers
   - Historic examples

# Transposition that doesn't hide much

## Internet meme (Cambridge research ???)

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoatnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

(" (sic)" deleted; "e" changed into "a"; comma inserted)

Source: `https://www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge/`

# Transposition that doesn't hide much

## Internet meme (Cambridge research ???)

According to a researcher at Cambridge University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be at the right place. The rest can be a total mess and you can still read it without problem. This is because the human mind does not read every letter by itself, but the word as a whole.

(" (sic)" deleted; "e" changed into "a"; comma inserted)

Source: `https://www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge/`

# Outline

1 Theoretic considerations
- Permutations

2 Framework for cryptography

3 Transpositions
- Geometric/Route ciphers
- Permutation ciphers
- Historic examples

# Generating transpositions (1): Skytale

# The Skytale (Scytale)

- Already used by the Spartan general Lysander
- Narrow parchment wound around a piece of wood of a given diameter
- Each side of the communication channel

  needs an identical piece of wood
- The length of wood is not important, as long as the message “fits”
- Encryption corresponds to a **simple columnar transposition**[3]
  - Decryption corresponds to a “row transposition”

[3] Some call this an example of a route transposition

# Simple columnar (route) transposition

$$\begin{pmatrix} 0 & 1 & \cdots & c-1 \\ c & c+1 & \cdots & 2c-1 \\ \hdotsfor{4} \\ (r-1)c & (r-1)c+1 & \cdots & rc-1 \end{pmatrix}$$

- r corresponds to the circumference (measured in letters) of the wooden stick
- The total length of the message is $rc$
- The plaintext is written in row by row
- The ciphertext is read out column by column

# Formulas relating a text string to the rectangular block

- From rectangle to string row by row

$$(i, j) \mapsto \; ic + j$$

- From string to rectangle row by row

$$i \mapsto \; ([i/c], i \pmod{c})$$

- From rectangle to string column by column

$$(i, j) \mapsto \; jr + i$$

- From string to rectangle column by column

$$i \mapsto \; (i \pmod{r}, [i/r])$$

# Generating transpositions (2): Railfence

- A railfence cipher has as key information only the depth of the fence
- Items are written down in a zigzag pattern going down and up (or up and down) the fence
- It can be described as an inefficient row transposition with holes (grille)
- Also, by zigzagging right and left, it can be described as a columnar transposition with holes

# Generating transpositions (3): Routes

There are many more ways to read from a rectangle

or for that matter any other geometric shape

**Exercise**

Match Colonel Parker Hitt's methods from Figure 3.2 in Holden's book

with the route ciphers from Figure 2.2 in Hans van der Meer's syllabus

# Outline

1 Theoretic considerations
- Permutations

2 Framework for cryptography

3 Transpositions
- Geometric/Route ciphers
- Permutation ciphers
- Historic examples

# Permutation(s of positions) ciphers

- Divide the plaintext into blocks of n letters and apply (the same) position permutation to each block separately
- The last block is padded with "nulls"
  - The permutation maps a ciphertext position to a plaintext position

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}$$

  - This permutation is the same as given by the following mapping

$$\begin{pmatrix} 4 & 1 & 3 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

  - **Warning: confusion!**
    In Math of Secrets this is also denoted by the string "4132"
- A string can be remembered by using a keyword, like "TALE" for the sequence 4132, using the alphabetic order

# Permutation cipher compositions (products)

- Suppose the block sizes are $p$ and $q$
- Then the block size of the product is given by
  - $\text{lcm}(p, q)$, the least common multiple of $p$ and $q$
- Hence in the case of $p = q$ you get nothing new

# Keyed columnar transpositions

- These are compositions (products) of permutations and routes
- They offer a better diffusion of the plaintext

  throughout the whole ciphertext
- There is still no confusion though
- Shannon argued that both confusion and diffusion are important
  - We will see later how modern block ciphers achieve both

# Outline

1 Theoretic considerations
- Permutations

2 Framework for cryptography

3 Transpositions
- Geometric/Route ciphers
- Permutation ciphers
- Historic examples

# A historic transposition

## Battle of Fredericksburg

Washington, D.C., November 25, 1862

To general Burnside, Falmouth, Virginia

Can Inn Ale me withe 2 oar our Ann pas Ann me flesh ends N. V. Corn Inn out with U cud Inn heaven day nest We roe Moore Tom darkey hat Greek Why Hawk of Abbott Inn B chewed I if.

From whom?

# Transpositions

Adapted from slides by Hans van der Meer

# Civil War

Message of president Lincoln to general major Burnside, dated Washington, November 25, 1862

BURNSIDE, Falmouth, Virginia
Can Inn Ale me withe 2 oar our Ann pas Ann me flesh ends N. V. Corn Inn out with U cud Inn heaven day nest We roe Moore Tom darkey hat Greek Why Hawk of Abbott Inn B chewed I if.

BURNSIDE, Falmouth, Virginia
If I should be in a boat off Aquia Creek at dark tomorrow, Wednesday evening, could you, without inconvenience, meet me and pass an hour or two with me? A. Lincoln

# Lincoln – June 1, 1863

PLAINTEXT For Colonel Ludlow. Richardson and Brown, correspondents of the Tribune, captured at Vicksburg, are detained at Richmond. Please ascertain why they are detained and get them off if you can. The President U.S.

CRYPTOGRAM guard adam them they at wayland brown for kissing venus correspondents at neptune are off nelly turning up can get why detained tribune and times richardson the are ascertain and you fills belly this if detained please odor of ludlow commissioner

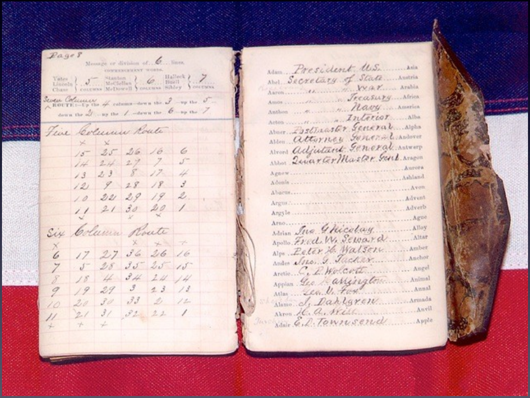
# Union codebook

indicator GUARD
 5x7 route tramp

null without meaning

filler on empty place

codewords
 adam = President US
 nelly = 4:30 pm
 neptune = Richmond
 odor = Vicksburg
 wayland = captured
 venus = colonel



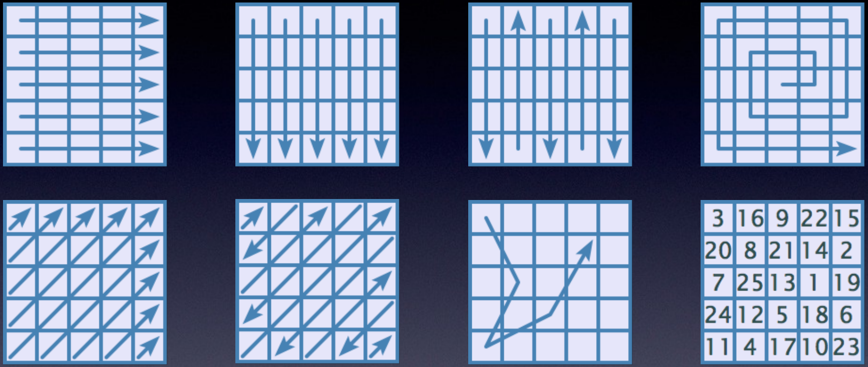
# Indicator GUARD

| | | | | |
|---|---|---|---|---|
| → | ↓ | STOP | ↓ | ← |
| kissing | ↓ | commissioner | ↓ | times |
| for | venus | Ludlow | Richardson | and |
| Brown | correspondents | of | the | Tribune |
| wayland | at | odor | are | detained |
| at | neptune | please | ascertain | why |
| they | are | detained | and | get |
| them | off | if | you | can |
| adam | nelly | this | fills | up |
| ↑ | turning | ↑ | belly | ↑ |
| ↑ | ↓ | ↑ | ← | ↑ |
| ↑ START ↑ | → | → | → | ↑ |

# Route transposition


plaintext: MAKKERS STAAKT UW WILD GERAAS
cryptogram: MRAWE ASKIR KSTLA KTUDA EAWGS

# Route transposition




- exotic routes are knight tour, magic square
- different routes for write-in and read-out
- spiral routes show readable fragments

# Column transposition

| M | U | S | K | U | S |
|---|---|---|---|---|---|
| 2 | 5 | 3 | 1 | 6 | 4 |
| D | E | V | R | A | G |
| E | N | V | A | N | H |
| E | T | T | E | N | T |
| A | M | E | N | Z | Y |
| N | N | O | G | G | E |
| H | E | I | M | X | X |

←transposition block filled completely

```
pt: DE VRAGEN VAN HET TENTAMEN ZIJN NOG GEHEIM XX
ct: RAENG MDEEA NHVVT EOIGH TYEXE NTMNE ANNZG X
```

Column division is unambiguous

```
RAENGM DEEANH VVTEOI GHTYEX ENTMNE ANNZGX
```

# Transposition key

| | | | | | |
|---|---|---|---|---|---|
| M | U | S | K | U | S |
| M | U | S | 1 | U | S |
| 2 | U | S | 1 | U | S |
| 2 | U | 3 | 1 | U | S |
| 2 | U | 3 | 1 | U | 4 |
| 2 | 5 | 3 | 1 | U | 4 |
| 2 | 5 | 3 | 1 | 6 | 4 |
| 2 | 5 | 3 | 1 | 6 | 4 |

Replace the letters of the key, one by one, by digits in alphabetic order

MUSKUS → 253164

# Column transposition

| M | U | S | K | U | S |
|---|---|---|---|---|---|
| 2 | 5 | 3 | 1 | 6 | 4 |
| D | E | V | R | A | G |
| E | N | V | A | N | H |
| E | T | T | E | N | T |
| A | M | E | N | Z | Y |
| N | N | O | G | G | E |
| H | E | I | M | | |

←transposition block **not** completely filled

```
pt: DE VRAGEN VAN HET TENTAMEN ZIJN NOG GEHEIM XX
ct: RAENG MDEEA NHVVT EOIGH TYEEN TMNEA NNZG
```

## Column division ambiguous

```
RAENGM DEEANH VVTEOI GHTYEE NTMNE ANNZG
RAENGM DEEANH VVTEOI GHTYE ENTMNE ANNZG
RAENG MDEEA NHVVTE OIGHTY EENTMN EANNZG
```

# Column transposition

| 2 | 1 | 4 | 7 | 3 | 6 | 5 |
|---|---|---|---|---|---|---|
| | A | A | | N | | V |
| A | | | | L | | |
| O | | P | P | E | | A |
| R | L | H | A | R | | B |
| O | U | R | D | O | O | R |
| J | A | P | A | N | | |

```
pt: AANVAL OP PEARL HARBOUR DOOR JAPAN
ct: ALUAA OROJN LERON APHRP VABRO PADA
```

Irregular block makes division even harder
- Japanse K1 around 1940 J19 encicode
- Zendia transpositions

# Dubbele transpositie

| H | A | R | I | N | G | T | O | N |
|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 8 | 4 | 5 | 2 | 9 | 7 | 6 |
| D | E | N | B | R | I | E | L | V |
| O | O | R | D | E | G | E | U | Z |
| E | N | G | E | V | A | L | L | E |
| N | A | L | V | A | N | A | A | R |
| M | A | D | R | I | D | | | |

| L | E | E | S | B | R | I | L |
|---|---|---|---|---|---|---|---|
| 5 | 2 | 3 | 8 | 1 | 7 | 4 | 6 |
| E | O | N | A | A | I | G | A |
| N | D | D | O | E | N | M | B |
| D | E | V | R | R | E | V | A |
| I | V | Z | E | R | L | U | L |
| A | N | R | G | L | D | E | E |
| L | A | | | | | | |

```
pt: DEN BRIEL VOOR DE GEUZEN GEVALLEN ALVA NAAR MADRID
ct: AERRL ODEVN ANDVZ RGMVU EENDI ALABA LEINE LDAOR EG
```

US Army Double Transposition

# Turning grille


0º 90º 180º 270º

```
pt: SIC ERGO ELEMENTIS
ct: NSLIR TCGIE OSMEE E
```

Also called a “Fleissner grille”
Oldest: Stadtholder Willem IV in 1745
Latest: German army in 1917

# Classical Cryptography

## Transposition cryptanalysis

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.1, 2020/02/14 14:48:33 UTC)

Monday, February 17, 2020

1. Keyed columnar transpositions
   - Completely filled rectangles
   - Partly filled rectangles

2. Multiple anagramming

# Outline

1. Keyed columnar transpositions
   - Completely filled rectangles
   - Partly filled rectangles

2. Multiple anagramming

# Outline

1 Keyed columnar transpositions
- Completely filled rectangles
- Partly filled rectangles

2 Multiple anagramming

# Determine rectangle width

- Guess a width (a divisor of the length) and fill in the rectangle as if it was a simple columnar transposition
- Look at the distribution of vowels and consonants over the rows
- In the English language 38% of letters is a vowel
- For N random choices out of this vowel distribution
  - The expected number of vowels is $0.38 \cdot N$
  - The expected variance is $0.38 \cdot 0.62 \cdot N$
- For N letters from English texts the variance should be lower

# Math of Secrets: 3.6 transposition

```
OHIVR SVAHT BLRHL HLBIT MBETM NOEIO
ITETK ROWTN ATHIG NSDEN UPBLN TSEMA
TADAA ERARI AOWSA YIAPT NAEOW BCDRE
WAHMT GEDER HFDDT EAEHA TEHME IELBO
HIUSI EKIUE UHESL MTKSE CREP
```

Calculate the variance of number of vowels for all the possible rectangles you can form from this block. Assume that the minimum width or length is 4.

# Anagramming

- After determination of the rectangle shape

  one can start anagramming the columns
- Look for probable digraphs for two columns
- Use the **contact method** for the most probable choice(s)
- Therefore calculate the **log weights** for the options
- The closer this sum of logarithms of digraph frequencies

  is to zero, the more probable the option is

# An Italian military example

RIOQK DEEFG ATCIE EGNEE NRMEN

NTOAV PTINT BAALL IUSUR OSNOE

NACGC YZATA MLALR ROKOI IA

# Completely filled columns

Adapted from slides by Hans van der Meer

# Filled columns

1. Find the size of the transposition block

As an example take 72 letters

```
1  block of size 72: 2x36 3x24 4x18 6x12 8x9
2 blocks of size 36: 2x18 3x12 4x9 6x6
3 blocks of size 24: 2x12 3x8 4x6
```

2. Divide the cryptogram in columns

3. Anagram these columns

# Example

Italian military message with

72 letters in one block

```
RIOQK DEEFG ATCIE EGNEE NRMEN
NTOAV PTINT BAALL IUSUR OSNOE
NACGC YZATA MLALR ROKOI IA
```

How many columns?

# Use vowel distribution

| | | |
|---|---|---|
| 2 RCNAOM | | |
| 4 IINAEL | | |
| 3 OETLNA | | |
| 3 QEOLAL | 4 RGEOAOCL | |
| 2 KGAICR | 5 IAEAASVR | 3 RFGNIIOZR |
| 1 DNVUGR | 1 OTNVLNZR | 4 IGNNNUEAR |
| 2 EEPSCC | 3 QCRPLOAO | 4 OAETTSNTO |
| 3 EETUVK | 3 KIMTIETK | 5 QTEOBUAAK |
| 2 FNIRZO | 6 DEEIUNAO | 3 KCNAARCMO |
| 3 GRNOAI | 4 EENNSAMI | 4 DIRVAOGLI |
| 2 AMTSTI | 3 EGNTUCLI | 4 EEMPLSCAI |
| 3 TEBNAA | 2 FNTBRGAA | 4 EEETLNVLA |
| 12X6 | 9X8 | 8X9 |

# Anagramming

Special combination of Q and U

| | |
|---|---|
| 123456789 | 2345789 16 |
| RFGNIIOZR | FGNIOZR RI |
| IGNNNUEAR | GNNNEAR IU |
| OAETTSNTO | AETTNTO OS |
| QTEOBUAAK | TEOBAAK QU |
| KCNAARCMO | CNAACMO KR |
| DIRVAOGLI | IRVAGLI DO |
| EEMPLSCAI | EMPLCAI ES |
| EEETLNVLA | EETLVLA EN |

# Find good third column

4 options

| | | | | | |
|---|---|---|---|---|---|
| 2345789 | 16 | 163 | 164 | 167 | 168 |
| FGNIOZR | RI | RIG | RIN | RIO | RIZ |
| GNNNEAR | IU | IUN | IUN | IUE | IUA |
| AETTNTO | OS | OSE | OST | OSN | OST |
| TEOBAAK | QU | QUE | QUO | QUA | QUA |
| CNAACMO | KR | KRN | KRA | KRC | KRM |
| IRVAGLI | DO | DOR | DOV | DOG | DOL |
| EMPLCAI | ES | ESM | ESP | ESC | ESA |
| EETLVLA | EN | ENE | ENT | ENV | ENL |

# Use a digram count

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2345789 | 16 | 163 | | 164 | | 167 | | 168 | |
| FGNIOZR | RI | RIG | 40 | RIN | 114 | RIO | 110 | RIZ | 5 |
| GNNNEAR | IU | IUN | 46 | IUN | 46 | IUE | 40 | IUA | 35 |
| AETTNTO | OS | OSE | 91 | OST | 94 | OSN | 0 | OST | 94 |
| TEOBAAK | QU | QUE | | QUO | | QUA | | QUA | |
| CNAACMO | KR | KRN | 16 | KRA | 124 | KRC | 24 | KRM | 16 |
| IRVAGLI | DO | DOR | 108 | DOV | 42 | DOG | 19 | DOL | 81 |
| EMPLCAI | ES | ESM | 11 | ESP | 16 | ESC | 35 | ESA | 30 |
| EETLVLA | EN | ENE | 110 | ENT | 124 | ENV | 5 | ENL | 3 |

# A nice fourth column

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 235789 | 164 | 1642 | | 1645 | | 1649 | |
| FGIOZR | RIN | RINF | 5 | RINI | 54 | RINR | 0 |
| GNNEAR | IUN | IUNG | 13 | IUNN | 24 | IUNR | 0 |
| AETNTO | OST | OSTA | 21 | OSTT | 56 | OSTO | 113 |
| TEBAAK | QUO | QUOT | 30 | QUOB | 8 | QUOK | 0 |
| CNACMO | KRA | KRAC | 54 | KRAA | 24 | KRAO | 13 |
| IRAGLI | DOV | DOVI | 38 | DOVA | 24 | DOVI | 38 |
| EMLCAI | ESP | ESPE | 67 | ESPL | 0 | ESPI | 27 |
| EELVLA | ENT | ENTE | 102 | ENTL | 0 | ENTA | 121 |

# Remaining columns

| | | | | |
|---|---|---|---|---|
| 35789 | 1642 | 35789 | 164279853 | SOLUTION |
| GIOZR | RINF | GIOZR | RINF | RINFORZIG |
| NNEAR | IUNG | NNEAR | IUNG | IUNGERANN |
| ETNTO | OSTA | ETNTO | OSTA | OSTANOTTE |
| EBAAK | QUOT | EBAAK | QUOT | QUOTAKABE |
| NACMO | KRAC | NACMO | KRAC | KRACCOMAN |
| RAGLI | DOVI | RAGLI | DOVI | DOVIGILAR |
| MLCAI | ESPE | MLCAI | ESPECIALMENTE | ESPECIALM |
| ELVLA | ENTE | ELVLA | ENTE | ENTEVALLE |

What does this final text mean?

# Outline

1. Keyed columnar transpositions
   - Completely filled rectangles
   - Partly filled rectangles

2. Multiple anagramming

# Disrupted columnar transposition

- Using an incompletely (partly) filled rectangle
- Common start or ending of messages helps
- This can help to determine long and short columns
- Long columns to the left, short ones to the right
- Now do simultaneous anagramming on each of the three parts

# Incompletely filled columns

Adapted from slides by Hans van der Meer

# Partly filled block

TSURC KNLCA PTEPE TLTTN EKCOI OAODH O

| | | | |
|---|---|---|---|
| | | *lssslsl* | |
| | | EL | |
| | | CPTK | |
| *plaintext* | *lllssss* | CAETCA | *sssslll* |
| ATTACKP | TKPTNOO | TKPTNOO | TCCELKA |
| OSTPONE | SNTLEID | SNTLEID | SKAPTCO |
| DUNTILT | ULETKOH | ULETKOH | UNPETOD |
| HREEOCL | RCPTCAO | RCPTCAO | RLTTNIH |
| OCK | CAE | CAE | EOO |

# Identical beginning

*cryptogram 1*

```
BNTSE ARKCL CETTN BITER ROTAE LTNNO
NNENO OTOKM SZTGN YITDK LANAE FTFSN
PGNPA RWOIA OFGTF CTOTD NINOE WXERF
ASIOS TIDRR RMMAO ARPAT OUTIO BIEOA
GAAPN EIK
```

*cryptogram 2*

```
BNTSE INDOT LCETS AFPLE RROMO ISOEN
NONST IIUTO KMFEY KPCYI TDVSI NTAEF
TFSTO NTNAR WOARO EEKTF CTTLT AEANO
EWXPV TITIO STTTF OCMMA OOSCA NROUT
IEELS OAGAA ABITR T
```

# Similarities

*cryptogram 1*

| | | |
|---|---|---|
| BNTSEARKC | LCETTNBIT | ERROTAELT |
| NNONNENOO | TOKMSZTGN | YITDKLAN |
| AEFTFSNPGNP | ARWOIAOFG | TFCTOTDNI |
| NOEWXERFAS | IOSTIDRRR | MMAOARPAT |
| OUTIOBIEO | AGAAPNEIK | |

*cryptogram 2*

| | | |
|---|---|---|
| BNTSEINDOT | LCETSAFPL | ERROMOISOE |
| NNONSTIIU | TOKMFEYKPC | YITDVSINT |
| AEFTFSTONTN | ARWOAROEEK | TFCTTLTAEA |
| NOEWXPVTIT | IOSTTTFOC | MMAOOSCANR |
| OUTIEELSO | AGAAABITRT | |

# Accidental coincidences

Identical parts of this cryptogram lined up

long = short + 1
This A disturbs the pattern (accidental hit)
A belongs at the end of the line before

| *cryptogram 1* | *cryptogram 2* |
|---|---|
| BNTSEARKC | BNTSEINDOT |
| LCETTNBIT | LCETSAFPL |
| ERROTAELT | ERROMOISOE |
| NNONNENOO | NNONSTIIU |
| TOKMSZTGN | TOKMFEYKPC |
| YITDKLAN | YITDVSINT |
| AEFTFSNPGNP | AEFTFSTONTN |
| ARWOIAOFG | ARWOAROEEK |
| TFCTOTDNI | TFCTTLTAEA |
| NOEWXERFAS | NOEWXPVTIT |
| IOSTIDRRR | IOSTTTFOC |
| MMAOARPAT | MMAOOSCANR |
| OUTIOBIEO | OUTIEELSO |
| AGAAPNEIK | AGAAABITRT |

# Permuted columns

*cryptogram 1*

```
BLENTYEATNIMOA
NCRNOIFRFOOMUG
TEROKTTWCESATA
STONMDFOTWTOIA
ETTNSKSIOXIAOP
ANAEZLNATEDRBN
RBENTAPODRRPIE
KILOGNGFNFRAEI
CTTONANGIARTOK
      P  S
```

*cryptogram 2*

```
BLENTYEATNIMOA
NCRNOIFRFOOMUG
TEROKTTWCESATA
STONMDFOTWTOIA
ESMSFVSATXTOEA
IAOTESTRLPTSEB
NFIIYIOOTVFCLI
DPSIKNNEATOAST
OLOUPTTEEICNOR
T E CANKAT R T
```

# Long columns left

| *cryptogram 1* | *cryptogram 2* |
| --- | --- |
| ENBLENTYATIMOA | BLENTYEATNIMOA |
| FONCRNOIRFOMUG | NCRNOIFRFOOMUG |
| TETEROKTWCSATA | TEROKTTWCESATA |
| FWSTONMDOTTOIA | STONMDFOTWTOIA |
| SXETTNSKIOIAOP | ESMSFVSATXTOEA |
| NEANAEZLATDRBN | IAOTESTRLPTSEB |
| PRRBENTAODRPIE | NFIIYIOOTVFCLI |
| GFKILOGNFNRAEI | DPSIKNNEATOAST |
| NACTTONAGIRTOK | OLOUPTTEEICNOR |
| PS | T E CANKAT R T |

# Short columns right

| *cryptogram 1* | *cryptogram 2* |
|---|---|
| ENBLENTYATIMOA | ENBETYATMALNIO |
| FONCRNOIRFOMUG | FONROIRFMGCNOU |
| TETEROKTWCSATA | TETRKTWCAAEOST |
| FWSTONMDOTTOIA | FWSOMDOTOATNTI |
| SXETTNSKIOIAOP | SXEMFVATOASSTE |
| NEANAEZLATDRBN | TPIOESRLSBATTE |
| PRRBENTAODRPIE | OVNIYIOTCIFIFL |
| GFKILOGNFNRAEI | NTDSKNEAATTPOS |
| NACTTONAGIRTOK | TIOOPTEENRLUCO |
| PS | NTTECAKART |

# Three parts

Simultaneous anagramming per part
ENEMY BATTALION...

*cryptogram 1*

```
ENBETYATMALNIO
FONROIRFMGCNOU
TETRKTWCAAEOST
FWSOMDOTOATNTI
SXETSKIOAPTNIO
NEAAZLATRNNEDB
PRRETAODPEBNRI
GFKLGNFNAIIORE
NACTNAGITKTORO
PS
```

*cryptogram 2*

```
ENBETYATMALNIO
FONROIRFMGCNOU
TETRKTWCAAEOST
FWSOMDOTOATNTI
SXEMFVATOASSTE
TPIOESRLSBATTE
OVNIYIOTCIFIFL
NTDSKNEAATTPOS
TIOOPTEENRLUCO
NTTECAKART
```

# The final result

| *cryptogram 1* | *cryptogram 2* |
|---|---|
| ENEMYBATTALION | ENEMYBATTALION |
| FORMINGFORCOUN | FORMINGFORCOUN |
| TERATTACKWESTO | TERATTACKWESTO |
| FWOODSATMOTTIN | FWOODSATMOTTIN |
| SXTAKEPOSITION | SXMOVEATFASTES |
| NEARLANTZANDBE | TPOSSIBLERATET |
| PREPAREDTOBRIN | OVICINITYOFFLI |
| GFLANKINGFIREO | NTSANDTAKETOSP |
| NATTACKINGTROO | TIONTOREPELCOU |
| PS | NTERATTACK |

# Outline

1 Keyed columnar transpositions
- Completely filled rectangles
- Partly filled rectangles

2 Multiple anagramming

# Multiple similarly encrypted texts

```
S E U I S M D M N A A S
J Y I N B N D H N O A L
L L N A A U E L C U I D
J E E I P K D C N A A E
B A I Y R D B D D U N G
```

- Getting multiple messages of the same length, encrypted with the same system, may come to the rescue
- Now you can try to use **multiple anagramming**

# Multiple Anagramming

Adapted from slides by Hans van der Meer

# Identical transpositions

From General Calamity
to Mayor Catastrophy

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | E | H | A | N | E | M | G | S | L | L | I | W | S | N | E | T | T | A | C | K | Y | E | I | A |
| 2 | A | E | B | P | S | O | U | R | P | E | M | O | C | E | E | T | U | N | R | I | S | T | E | R | S |
| 3 | A | F | O | E | T | O | R | T | D | A | E | R | T | E | F | D | I | N | C | A | S | E | R | E | T |
| 4 | T | U | O | P | W | A | R | U | R | E | F | F | O | Y | A | E | E | D | F | O | R | D | R | C | R |

Military terminology
Four parts with identical transposition
Simultaneous anagramming

# Probable word

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | E | H | A | N | E | M | G | S | L | L | I | W | S | N | E | T | T | A | C | K | Y | E | I | A |
| 2 | A | E | B | P | S | O | U | R | P | E | M | O | C | E | E | T | U | N | R | I | S | T | E | R | S |
| 3 | A | F | O | E | T | O | R | T | D | A | E | R | T | E | F | D | I | N | C | A | S | E | R | E | T |
| 4 | T | U | O | P | W | A | R | U | R | E | F | F | O | Y | A | E | E | D | F | O | R | D | R | C | R |

## ATTACK ENEMY

| 1 | A | | | T | | | T | | | A | | | C | K |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | P | R | S | A | U | N | A | U | N | P | R | S | I | S |
| 3 | E | C | T | A | I | N | A | I | N | E | C | T | A | S |
| 4 | P | F | R | T | E | D | T | E | D | P | F | R | O | R |

| 1 | E | | | | N | | E | | | | M | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | E | O | T | E | S | E | E | O | T | E | U | T |
| 3 | F | O | D | R | T | F | F | O | D | R | R | E |
| 4 | U | A | E | R | W | A | U | A | E | R | R | D |

# Simultaneous text

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | E | H | A | N | E | | G | S | L | L | I | W | S | N | E | T | T | A | | | E | I | A |
| 2 | A | E | B | P | S | O | | R | P | E | M | O | C | E | E | T | U | N | R | | | E | R | S |
| 3 | A | F | O | E | T | O | | T | D | A | E | R | T | E | F | D | I | N | C | | | R | E | T |
| 4 | T | U | O | P | W | A | | U | R | E | F | F | O | Y | A | E | E | D | F | | | R | C | R |

2 SUNRISE

4 REWARD

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A | | | T | | | T | | | A | | | C | K |
| 2 | P | R | S | A | U | N | A | U | N | P | R | S | I | S |
| 3 | E | C | T | A | I | N | A | I | N | E | C | T | A | S |
| 4 | P | F | R | T | E | D | T | E | D | P | F | R | O | R |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | E | | | | N | | E | | | | M | Y |
| 2 | E | O | T | E | S | E | E | O | T | E | U | T |
| 3 | F | O | D | R | T | F | F | O | D | R | R | E |
| 4 | U | A | E | R | W | A | U | A | E | R | R | D |

# Chosen letters

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | E | H | A | | | | G | S | L | L | I | W | S | N | | | | | | | | E | I | |
| 2 | A | E | B | P | | | | R | P | E | M | O | C | E | E | | | | | | | | E | R | |
| 3 | A | F | O | E | | | | T | D | A | E | R | T | E | F | | | | | | | | R | E | |
| 4 | T | U | O | P | | | | U | R | E | F | F | O | Y | A | | | | | | | | R | C | |

## 3 IN CASE OF

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A | | | T | | | T | | | A | | | C | K |
| 2 | | | S | | U | | | | N | | R | | I | S |
| 3 | | | T | | I | | | | N | | C | | A | S |
| 4 | | | R | | E | | | | D | | F | | O | R |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | E | | | | N | | E | | | | M | Y |
| 2 | | | T | | S | | | O | | | U | T |
| 3 | | | D | | T | | | O | | | R | E |
| 4 | | | E | | W | | | A | | | R | D |

# Extend

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | E | H | A | | | | G | S | L | L | I | W | S | N | | | | | | | | E | I | |
| 2 | A | E | B | P | | | | R | P | E | M | O | C | E | E | | | | | | | | E | R | |
| 3 | A | F | O | E | | | | T | D | A | E | R | T | E | F | | | | | | | | R | E | |
| 4 | T | U | O | P | | | | U | R | E | F | F | O | Y | A | | | | | | | | R | C | |

2 SUNRISE
3 IN CASE OF

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A | T | T | A | C | K | S | H | E | N |
| 2 | S | U | N | R | I | S | E | B | E | E |
| 3 | T | I | N | C | A | S | E | O | F | |
| 4 | R | E | D | F | O | R | Y | O | U | A |

# Another probable word

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | | | A | | | | G | S | L | L | I | W | | N | | | | | | | | E | I | |
| 2 | A | | | P | | | | R | P | E | M | O | C | | E | | | | | | | | E | R | |
| 3 | A | | | E | | | | T | D | A | E | R | T | | F | | | | | | | | R | E | |
| 4 | T | | | P | | | | U | R | E | F | F | O | | A | | | | | | | | R | C | |

## 3 DEFEAT

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | S | A | L | I | N | A | L | I | T | L | G | W |
| 2 | P | P | M | R | E | P | M | R | A | E | R | C |
| 3 | D | | E | | F | | E | | A | | T | |
| 4 | R | P | F | C | A | P | F | C | T | E | U | O |

# New word visible

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | | | A | | | | G | | L | L | I | W | | | | | | | | | | E | I | |
| 2 | A | | | P | | | | R | | E | M | O | C | | | | | | | | | | E | R | |
| 3 | A | | | E | | | | T | | A | E | R | T | | | | | | | | | | R | E | |
| 4 | T | | | P | | | | U | | E | F | F | O | | | | | | | | | | R | C | |

2 PREPARE

<table>
<tr><td>1</td><td>S</td><td>A</td><td>L</td><td>I</td><td>N</td><td>A</td><td>L</td><td>I</td><td>T</td><td>L</td><td>G</td><td>W</td></tr>
<tr><td>2</td><td>P</td><td>P</td><td>M</td><td>R</td><td>E</td><td>P</td><td>M</td><td>R</td><td>A</td><td>E</td><td>R</td><td>C</td></tr>
<tr><td>3</td><td>D</td><td colspan="3">E</td><td>F</td><td colspan="3">E</td><td colspan="2">A</td><td colspan="2">T</td></tr>
<tr><td>4</td><td>R</td><td>P</td><td>F</td><td>C</td><td>A</td><td>P</td><td>F</td><td>C</td><td>T</td><td>E</td><td>U</td><td>O</td></tr>
</table>

# Fragments

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | L | L | I | W | | | | | | | | | | E | | |
| 2 | | | | | | | | | | E | M | O | C | | | | | | | | | | E | | |
| 3 | | | | | | | | | | A | E | R | T | | | | | | | | | | R | | |
| 4 | | | | | | | | | | E | F | F | O | | | | | | | | | | R | | |

ENEMY WILL ATTACK

| | | | | | |
|---|---|---|---|---|---|
| 1 | E | N | E | M | Y |
| 2 | T | S | O | U | T |
| 3 | D | T | O | R | E |
| 4 | E | W | A | R | D |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | A | T | T | A | C | K | S | H | E |
| 2 | S | U | N | R | I | S | E | B | E |
| 3 | T | I | N | C | A | S | E | O | F |
| 4 | R | E | D | F | O | R | Y | O | U |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | S | I | N | A | T | G |
| 2 | P | R | E | P | A | R |
| 3 | D | E | F | E | A | T |
| 4 | R | C | A | P | T | U |

# Final steps

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | E | N | E | M | Y | W | I | L | | L | | A | T | T | A | C | K |
| 2 | T | S | O | U | T | C | O | E | M | E | M | S | U | N | R | I | S |
| 3 | D | T | O | R | E | T | R | A | E | A | E | T | I | N | C | A | S |
| 4 | E | W | A | R | D | O | F | E | F | E | F | R | E | D | F | O | R |

## Plaintext

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E | N | E | M | Y | W | I | L | L | A | T | T | A | C | K | S | H | E | S | I | N | A | T | G | E |
| T | S | O | U | T | C | O | M | E | S | U | N | R | I | S | E | B | E | P | R | E | P | A | R | E |
| D | T | O | R | E | T | R | E | A | T | I | N | C | A | S | E | O | F | D | E | F | E | A | T | R |
| E | W | A | R | D | O | F | F | E | R | E | D | F | O | R | Y | O | U | R | C | A | P | T | U | R |

# Classical Cryptography

## Encodings and digital ciphers

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.1, 2020/02/18 14:19:34 UTC)

Thursday, February 20, 2020

1. Codes and ciphers
2. Public codes
3. From public codes to ciphers
4. Digital ciphers

# Outline


1. Codes and ciphers
2. Public codes
3. From public codes to ciphers
4. Digital ciphers

# Codes and codebooks

- A **code** operates on **semantic** components
  - Words, paragraphs, ...
- A **codebook** is a **lookup table** for codes
- Codes can be very hard to cryptanalyze
- Possible analysis methods
  - Compare many different coded texts
  - Use side channels (other available sources)
  - Try to identify cribs
  - Build up knowledge over time

# An example codebook

Source: Slides Hans van der Meer

# Encodings

- An **encoding** is a transformation of pieces of information into another representation for communication or storage
- An encoding is keyless
- An encoding can be public or secret
- The pieces of information need not have a semantic value and can be single letters or symbols

# Ciphers and algorithms

- A **cipher** operates on meaningless components
    - Individual letters
    - Small groups of letters
    - Bits
    - Bytes
- Ciphers are **syntax** related
- Ciphers use **algorithms**
    - with secret (or public) keys as parameters
- Encryption/decryption is the process of applying/reversing a cipher

# Outline


1. Codes and ciphers
2. Public codes
3. From public codes to ciphers
4. Digital ciphers

# Polygraphic versus polyliteral ciphers/encodings

- Polygraphic ciphers/encodings translate a block of letters into another block of letters, numbers or symbols
  - An example is Porta's digraph system
- Polyliteral ciphers/encodings translate a single letter into a (larger, full) block of letters, numbers or symbols
  - Polyliteral ciphers/encodings are nothing more than a simple substitution into an "unknown", "bigger", but also "structured" alphabet which can henceforth be fractionated

# Polybius Square

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | A | B | C | D | E |
| 2 | F | G | H | IJ | K |
| 3 | L | M | N | O | P |
| 4 | Q | R | S | T | U |
| 5 | V | W | X | Y | Z |

Figure 1: A simple polyliteral[1] encoding (**Polybius**)

[1] Because we use digits this is also called a dinome substitution

# A rectangular variant

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | | A | B | C | D | E | F | G | H |
| 1 | I | J | K | L | M | N | O | P | Q |
| 2 | R | S | T | U | V | W | X | Y | Z |

Figure 2: A 0-based rectangular encoding for the full alphabet

But note it still uses the legacy A=01 encoding

# The standard legacy encoding

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | A | B | C | D | E | F | G | H | I |
| 1 | J | K | L | M | N | O | P | Q | R | S |
| 2 | T | U | V | W | X | Y | Z | | | |

Figure 3: A 0-based encoding for the full alphabet, with open space

- This encoding is just the regular legacy encoding translating A, ..., Z to 01[2], ..., 26
- 00, 27, 28 and 29 are available for more symbols if needed

[2] One may or may not remove leading 0s

# The standard modern encoding

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | A | B | C | D | E | F | G | H | I | J |
| 1 | K | L | M | N | O | P | Q | R | S | T |
| 2 | U | V | W | X | Y | Z | | | | |

Figure 4: A 0-based encoding for the full alphabet, with open space

This encoding is just the regular modern encoding

translating A, ..., Z to 00, ..., 25

# A table for every base b numeral system (1)

Let us for instance look at base $b = 3$

| | 00 | 01 | 02 | 10 | 11 | 12 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ␣ | A | B | C | D | E | F | G | H |
| 1 | I | J | K | L | M | N | O | P | Q |
| 2 | R | S | T | U | V | W | X | Y | Z |

Figure 5: A ternary encoding for the full alphabet including a space

It would have been so nice[3] to build computers based on the (balanced) ternary system instead of the usual binary one...

[3]The Russians tried to do so: `https://en.wikipedia.org/wiki/Setun`

# A table for every base b numeral system (2)

Let us now look at the common binary base $b = 2$

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 00 | | A | B | C | D | E | F | G |
| 01 | H | I | J | K | L | M | N | O |
| 10 | P | Q | R | S | T | U | V | W |
| 11 | X | Y | Z | | | | | |

Figure 6: A binary legacy encoding with room for $2^5 = 32$ symbols

Base32 is a modern variant with added symbols 2, 3, 4, 5, 6, 7

# The Bacon code (steganography)

- Francis Bacon (1561–1626)
- First use a binary code with a=0 and b=1
  - In the original we had I=J and U=V, coding 24 letters with A=aaaaa, ..., Z=babbb
  - In modern variants the full alphabet is encoded[4] with A=aaaaa, ..., Z=bbaab
- SEconDIY HIDE The INDivIDuAL BitS by USiNg GLypH PROPeRTieS LiKE ColOR, ITaLIZatIon, SIze, ...

[4]Holden's book uses A=aaaab, ..., Z=bbaba

# The Teletypewriter

- Émile Baudot (1845–1903)
  - Baudot code
  - Paper tape with punched holes
  - 5 positions or bits
- Gilbert Vernam (1890–1960)
  - Secures Baudot code transmission
  - Uses a second (key)tape to be XORed with the plaintext tape
  - Essentially creating a one-time pad

# The wonderfully versatile XOR

- XOR is a binary (bitwise) operation
  - Its nice properties derive from addition modulo 2
  - Modulo 2 subtraction is the same as addition
  - Encryption works by $c = p \oplus k$
  - and since $k \oplus k = 0$
  - decryption works by $p = c \oplus k$
- XOR also has a ternary, quaternary, ... variant
  - Multiple inputs and one output
  - Can be combined in arbitrary trees
    - And with some care even in graphs with loops

# Outline


1. Codes and ciphers
2. Public codes
3. From public codes to ciphers
4. Digital ciphers

# Length tricks

- Nulls
  - Using encoding symbols with no corresponding plaintext
- Straddling (“with a leg on each side”)
  - Use different length encoding strings for different plaintext letters
  - Usually the frequently occurring letters use a smaller length
  - This will result in compression properties

# The straddling checkerboard (1)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | A | B | C | D | E | F | G |
| 1 | H | I | J | K | L | M | N | O | P | Q |
| 2 | R | S | T | U | V | W | X | Y | Z | |

Figure 7: Why are the first three positions blank?

# The straddling checkerboard (2)

| | 0 | 1 | 8 | 3 | 4 | 5 | 2 | 9 | 7 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | T | R | E | A | S | O | N |
| 0 | B | C | D | F | G | H | I | J | K | L |
| 1 | M | P | Q | U | V | W | X | Y | Z | . |
| 8 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Figure 8: A variant that compresses (most occurring letters monome)

Source: slides Hans van der Meer

# The straddling checkerboard (3)

| | 0 | 1 | 8 | 3 | 4 |
|---|---|---|---|---|---|
| | A | E | I | O | U |
| 5 | B | C | D | F | G |
| 2 | H | K | L | M | N |
| 9 | P | Q | R | S | T |
| 7 | V | W | X | Y | Z |

Figure 9: A variant where the 6 can be used as a null

Source: slides Hans van der Meer

# The straddling checkerboard (4)

| | 0 | 1 | 8 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 2 | A | B | C | D | E | F |
| 9 | G | H | I | J | K | L |
| 7 | M | N | O | P | Q | R |
| 62 | S | T | U | V | W | X |
| 67 | Y | Z | 0 | 1 | 2 | 3 |
| 69 | 4 | 5 | 6 | 7 | 8 | 9 |

Figure 10: A dinome-trinome variant

Source: slides Hans van der Meer

# The straddling checkerboard (5)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Q | R | S | T | U |
| | | | V | W | X | Y | Z |
| | | | E | T | N | R | O |
| L | F | A | A | B | C | D | F |
| M | G | B | G | H | I | J | K |
| N | H | C | L | M | P | Q | S |
| O | I | D | U | V | W | X | Y |
| P | K | E | Z | . | $ | ( | ) |

Figure 11: Lots of homophones

Source: slides Hans van der Meer

# Cryptanalysis of straddling checkerboards

- Identify dinome coordinates
  - They occur more frequently
  - They have lots of different contacts
  - Look at repetition of four or more identical digits
  - Look at patterns like `abab`
- Solve the resulting monoalphabetic substitution
- And possibly identify the key used

# Fractionation after polyliteral encoding

- After having encoded letters one may consider subunits of polyliterals
  - In the binary case those subunits could be bits
- More substitutions and especially transpositions can be executed
  - That is what classic and modern block ciphers like DES and AES do
- The resulting new subunits might be assembled again into polyliterals
  - Which can then possibly be translated back to the original alphabet

# Fractionating system example: ADFGVX (1)

| | A | D | F | G | V | X |
|---|---|---|---|---|---|---|
| A | b | 5 | x | q | j | c |
| D | 6 | y | r | k | d | 7 |
| F | z | s | l | e | 8 | 1 |
| G | t | m | f | 9 | 2 | u |
| V | n | g | 0 | 3 | v | o |
| X | h | a | 4 | w | p | i |

Figure 12: ADFGVX 6-by-6 square

# Fractionating system example: ADFGVX (2)

- First use the polyliteral ADFGVX square
- Then use a keyed columnar transposition
- Example encryption with keyword GANDHI and square filled as in previous slide
  - AGGAV AXGDA DFGGA FXFFV VXXFG XXVGF VAAXX ADAXG FFFFVD
- Exercise: decode this message

# Outline


1. Codes and ciphers

2. Public codes

3. From public codes to ciphers

4. Digital ciphers

# Shannon's theory (1)

- **Confusion**
  - Each ciphertext bit has complex (nonlinear) relations with the plaintext and key bits
  - Mostly done by substitutions
- **Diffusion**
  - Each plaintext or key bit affects many bits of the ciphertext
  - Mostly done by transpositions

# Shannon's theory (2)

- **Mixing** transformation (function) H
  - Non-secret, confusing and diffusing transformation
  - A transposition (T), followed by an alternation of linear (L) maps and substitutions (S)
  - $H = L \circ S \circ L \circ S \circ L \circ T$
  - Both T and L operate on full blocks of letters
  - S operates componentwise, on each individual letter

# Shannon's theory (3)

- Shannon's cipher construction
  - Uses one, two or even more mixing transformations
    - For two mixings this is $C = T_k \circ H_2 \circ S_j \circ H_1 \circ R_i$
    - $i, j, k$ is keying material for simple ciphers $R, S, T$
    - Here secret keys enter the scene by adding more confusion, typically through the simple substitutions R, S and T

# SP-networks

- **SP-networks** resemble Shannon's construction
  - Works with bits instead of larger alphabets
- Uses **large diffusing transpositions** of bits
- Uses **smaller confusing polygraphic substitutions** of sequences of bits (bytes, nibbles, ...)
- Alternates these in a number of rounds
- Mixes in (parts of) the key at the start of each round
  - Mixing uses simple XORs
  - Also at the end the key is once more mixed in

# Feistel networks (building block)

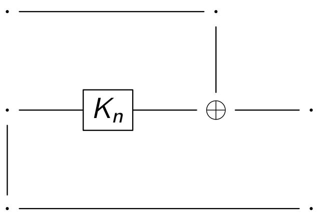



Figure 13: Building block (also used upside down)

# Feistel networks (first few steps)

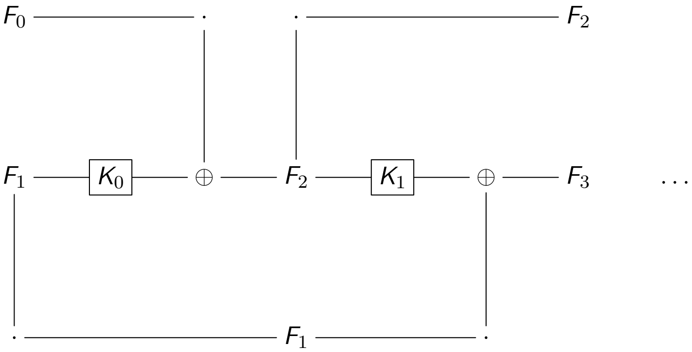



Figure 14: $F_2 = \mathcal{F}(K_0, F_1) \oplus F_0; F_3 = \mathcal{F}(K_1, F_2) \oplus F_1$

# Feistel network encryption sequence

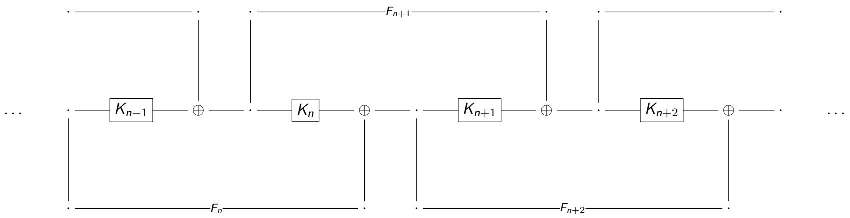



Figure 15: $F_{n+2} = \mathcal{F}(K_n, F_{n+1}) \oplus F_n$

# Simpler Feistel network building block

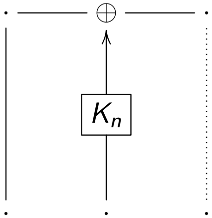



Figure 16: Building block (also used upside down)

# Simpler Feistel network first steps

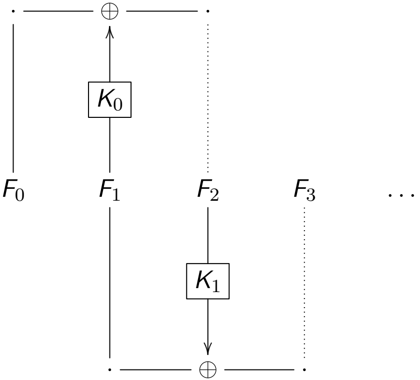



Figure 17: $F_2 = \mathcal{F}(K_0, F_1) \oplus F_0; F_3 = \mathcal{F}(K_1, F_2) \oplus F_1$

# Simpler Feistel network encryption sequence

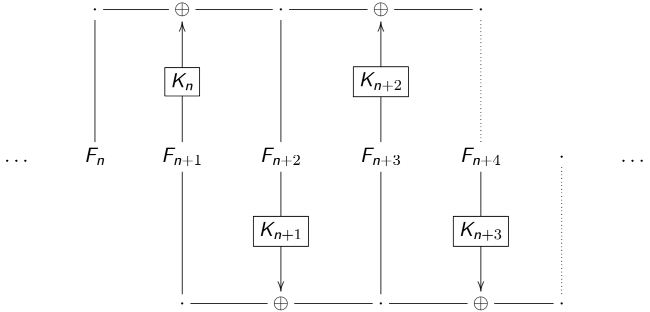



Figure 18: $F_{n+2} = \mathcal{F}(K_n, F_{n+1}) \oplus F_n$

# Enigma and the U-Boat War in the Atlantic Ocean

## Strategy – Tactics – Intelligence

*dr. Hans van der Meer*

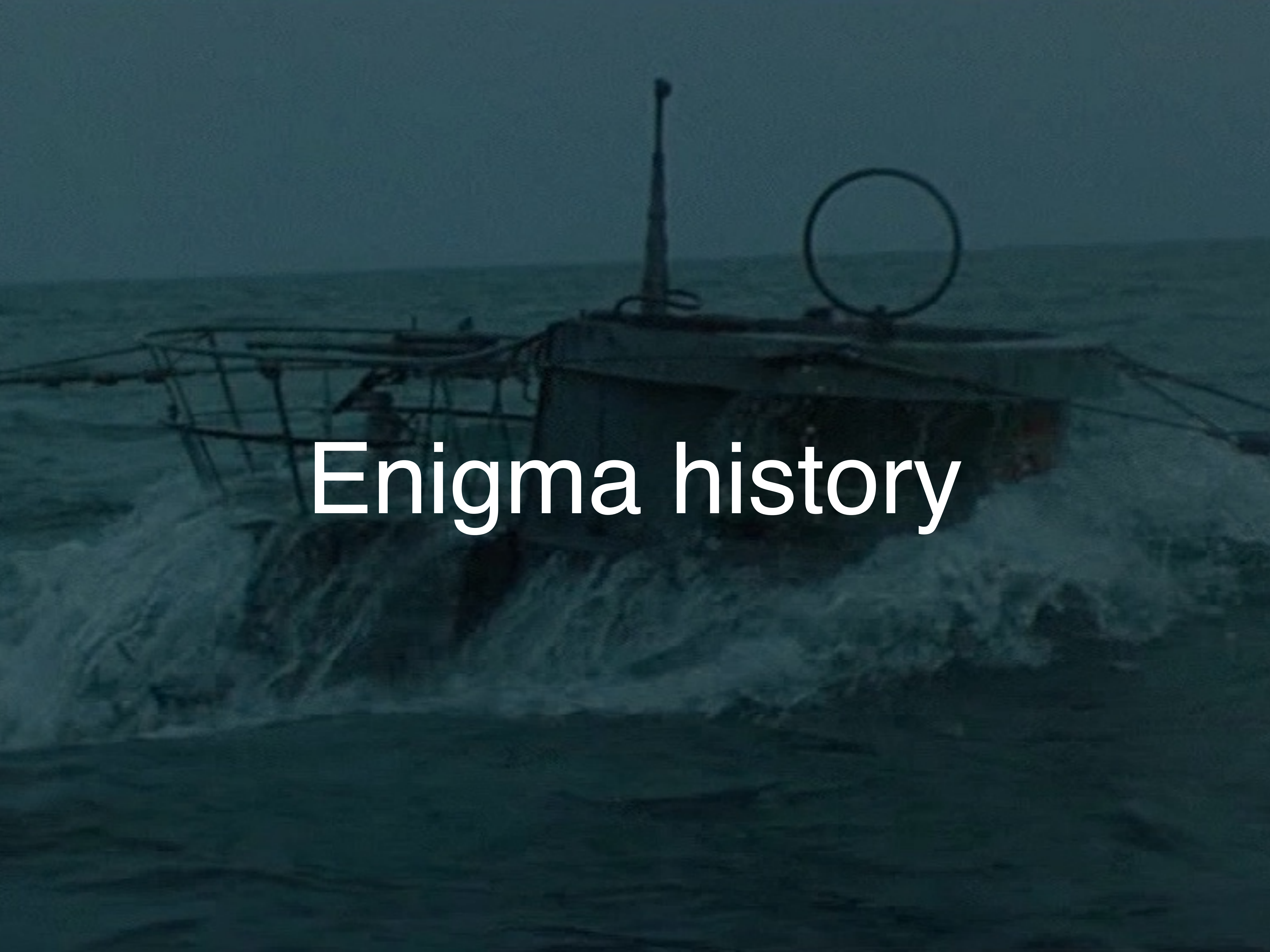
*17-02-2020*

◉ **Strategy** – Germany cuts British supply lines
◉ **Tactics** – Attack and defense on shipping lanes
◉ **Intelligence** – Breaking each others codes

# Outline

- Enigma history
- Cryptography of the Enigma
- Polish mathematicians break Enigma
- British mechanisation of breaking Enigma
- Befehlshaber der U-Boote vs Western Approaches
- Aftermath

# Enigma history



# Development

1915 Spengler and Van Hengel

1919 Koch patent

1920 Chiffriermaschinen AG

1923 Enigma-A

1924 Kriegsmarine Funkschlüssel-C

1927 Enigma in Reichswehr

1932 Enigma everywhere

1941 Kriegsmarine M4

Van Hengel

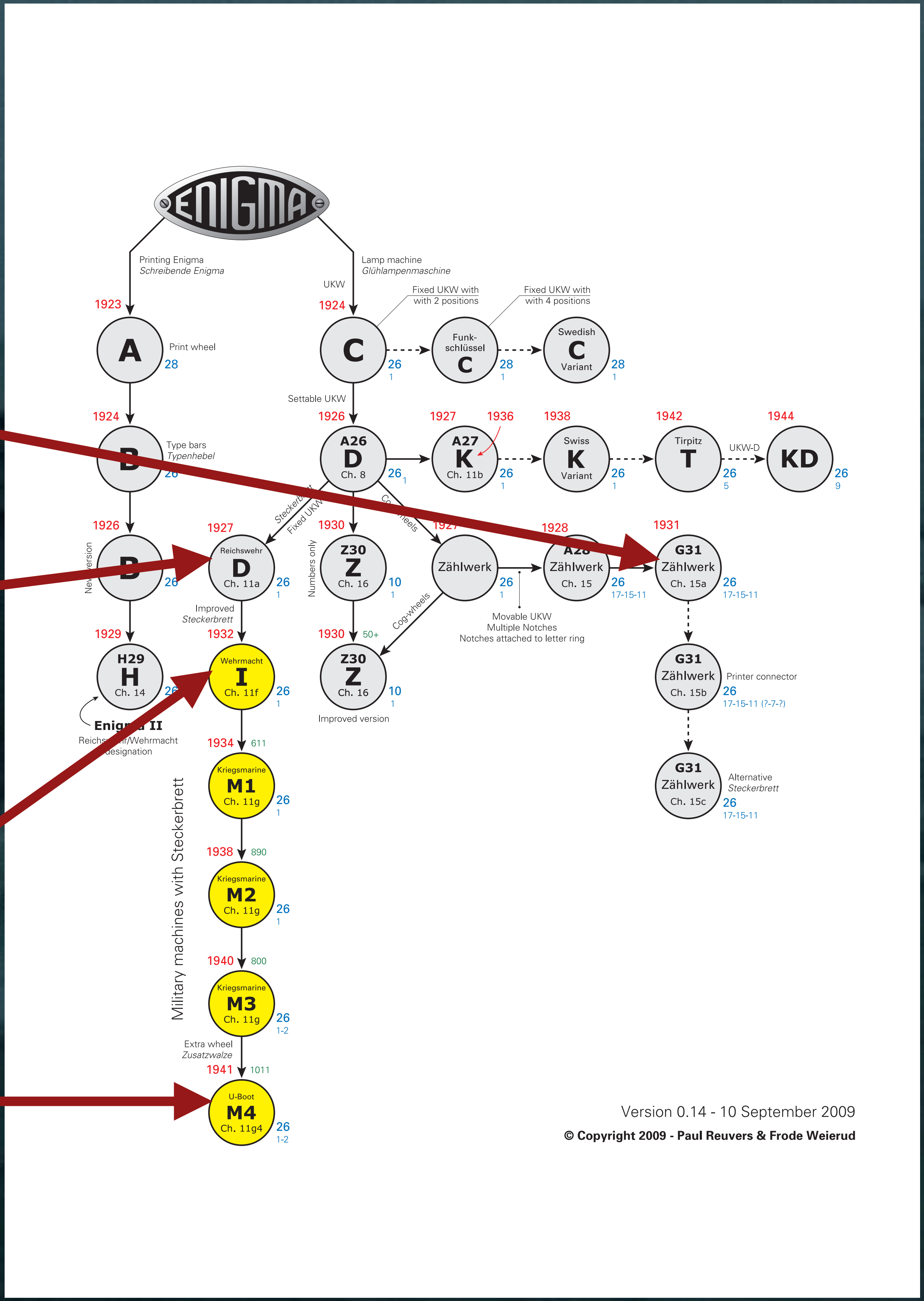
# Enigma family tree

Abwehr, Dutch navy

Reichswehr 1927

Wehrmacht 1932 plugboard

M4 in U-boats since Feb1942



# Some pictures

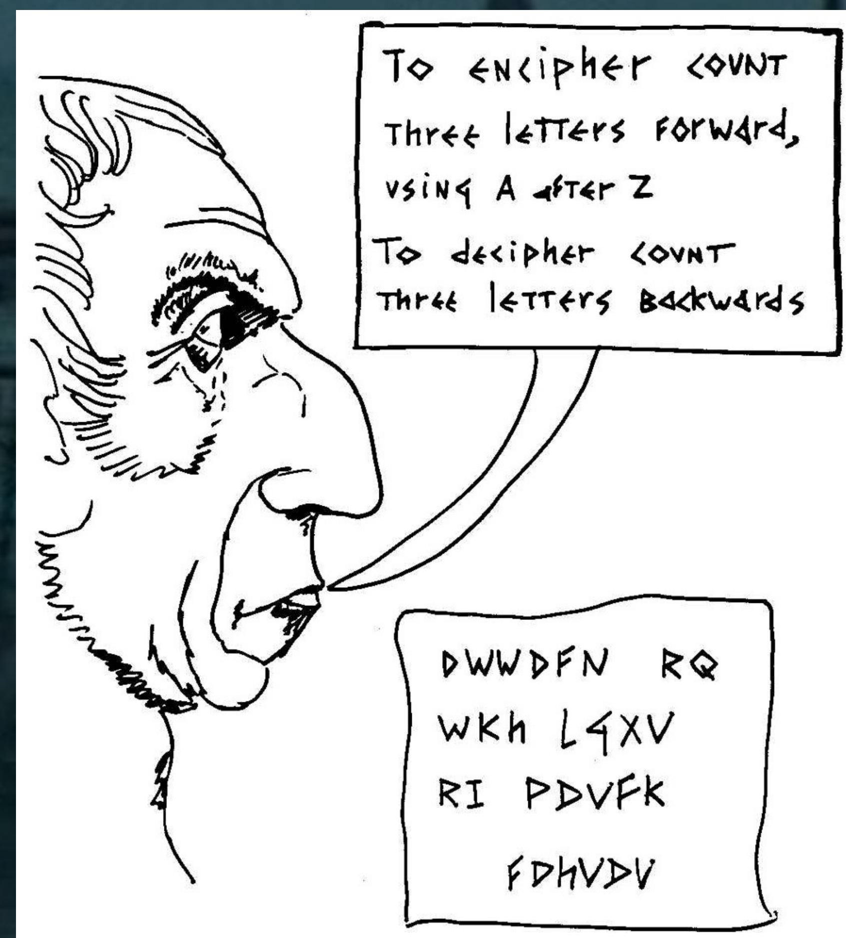
Enigma - G
Abwehr


Enigma
Wehrmacht

Enigma - M4
Kriegsmarine


Cryptography

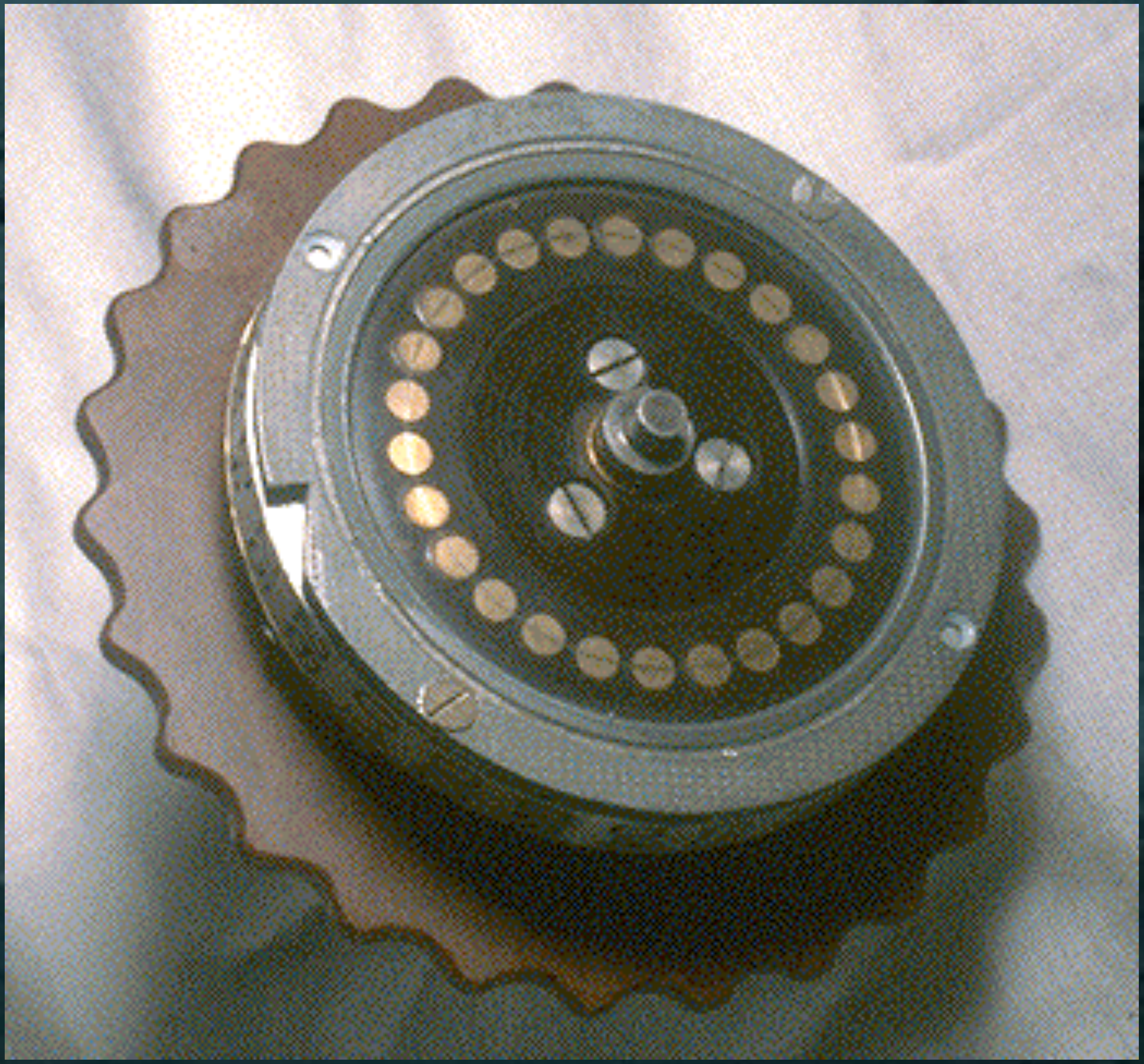
# Caesar encryption



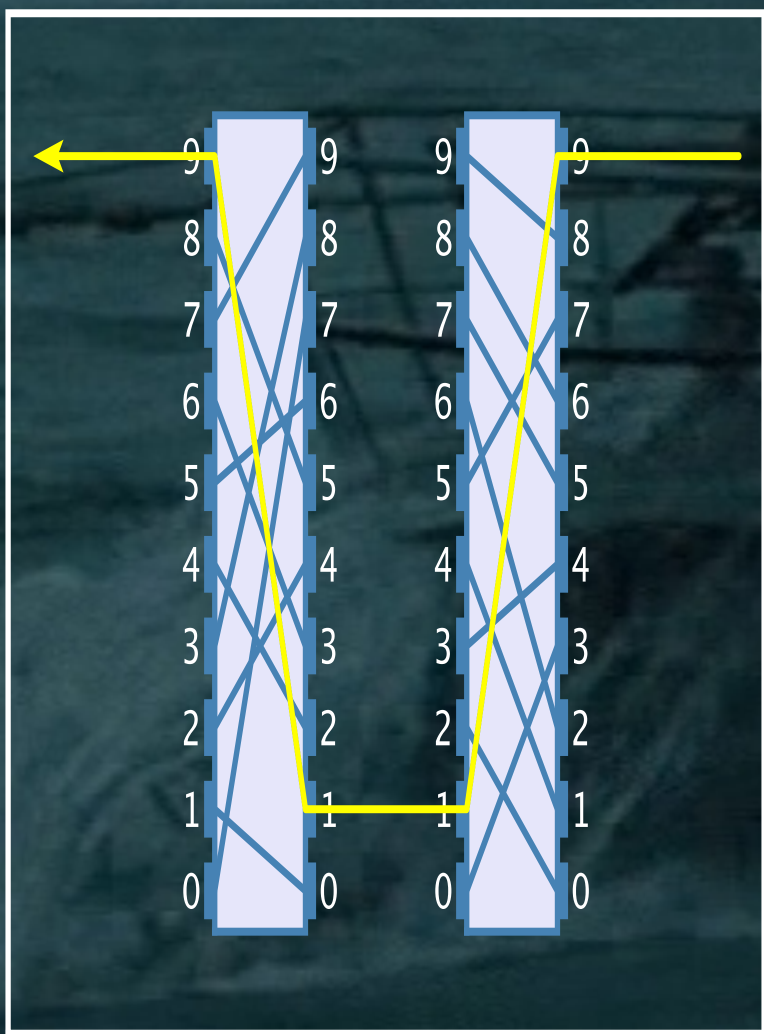

ATTACK ON THE IDUS OF MARCH CAESAR

DWWDFN RQ WKH LGXV RI PDVFK FDHVDV

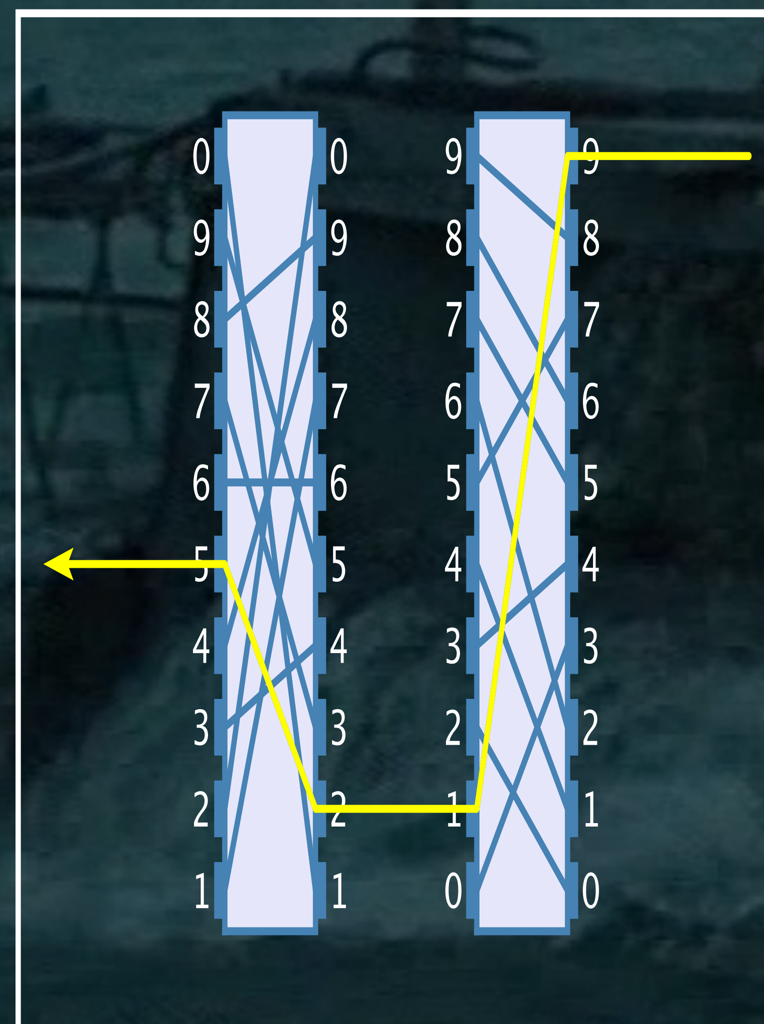
# Enigma rotor



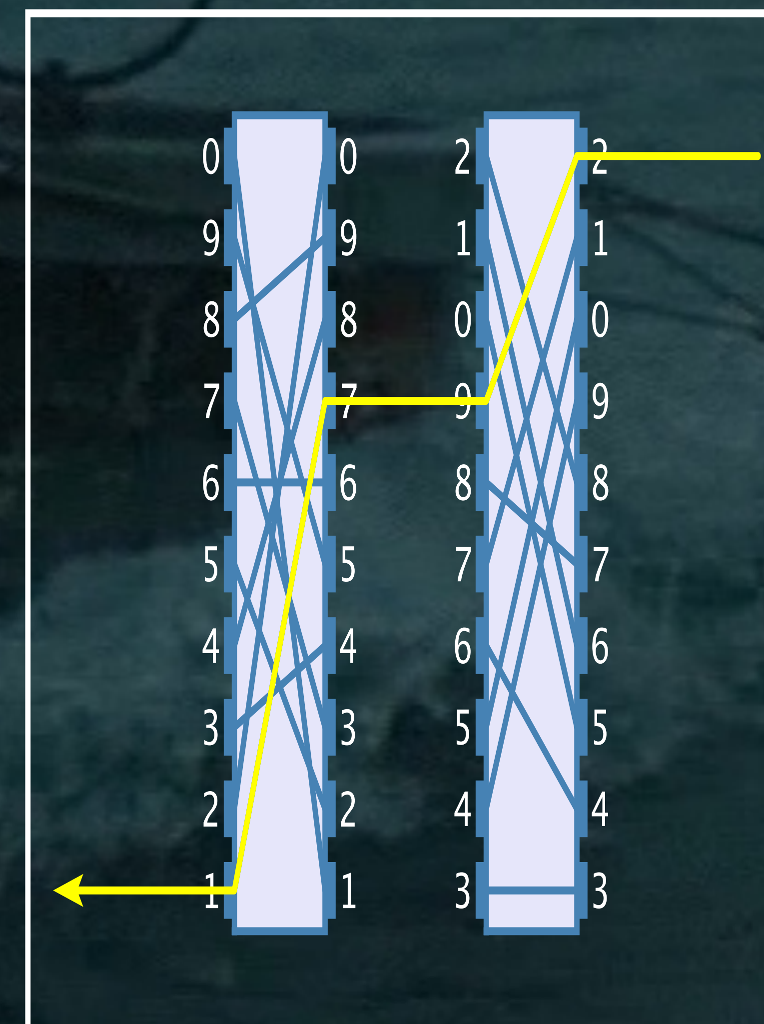
# Coupled rotors




current flow

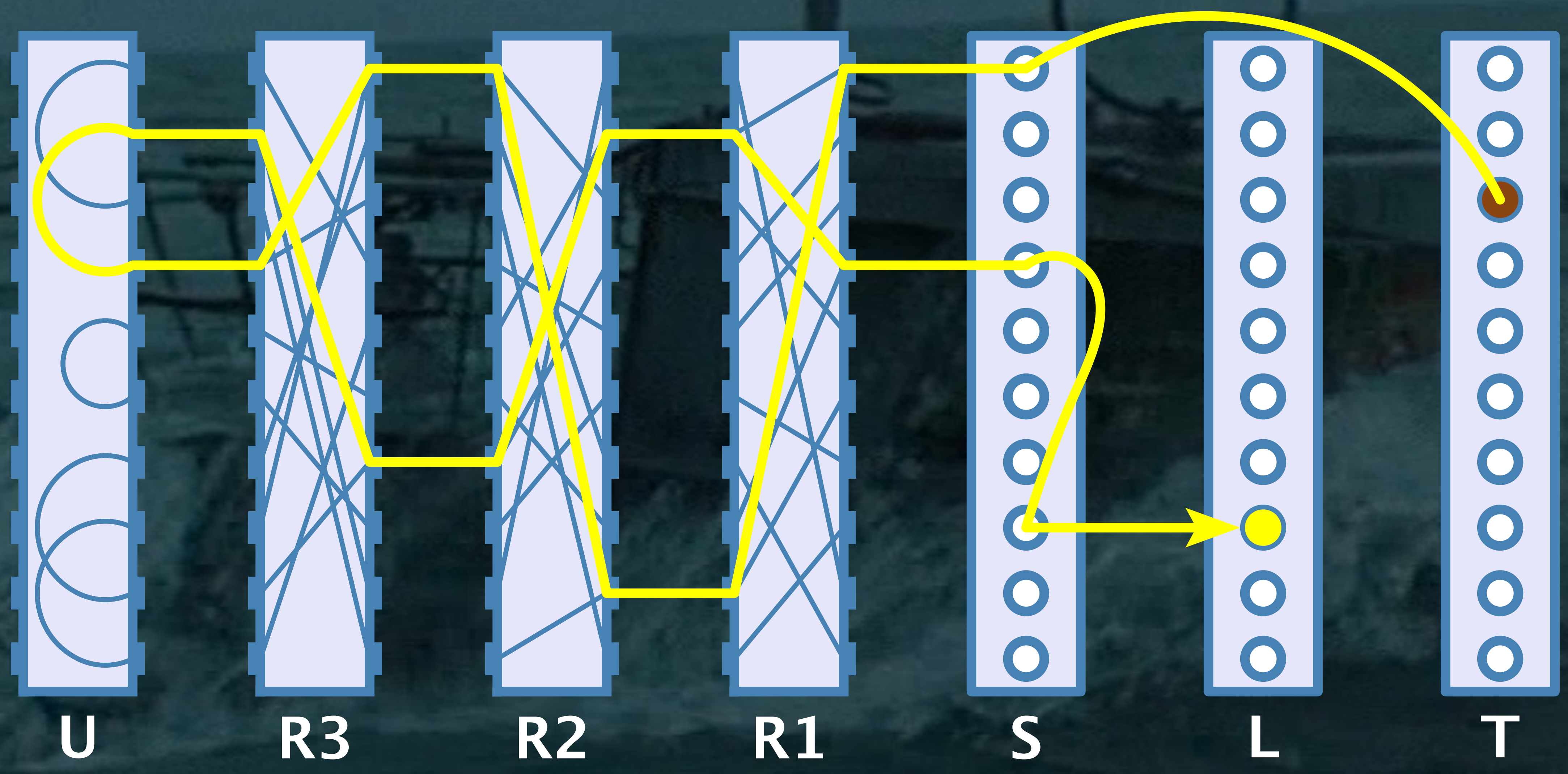



left 1 step



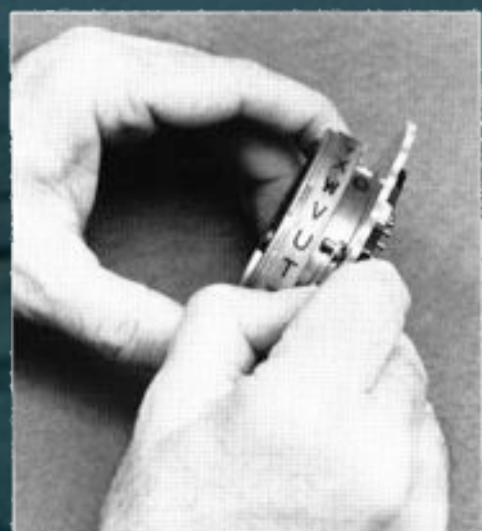

right 3 steps

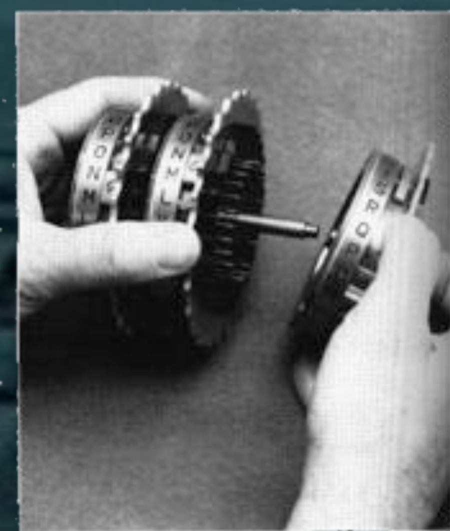
# Enigma with plugboard

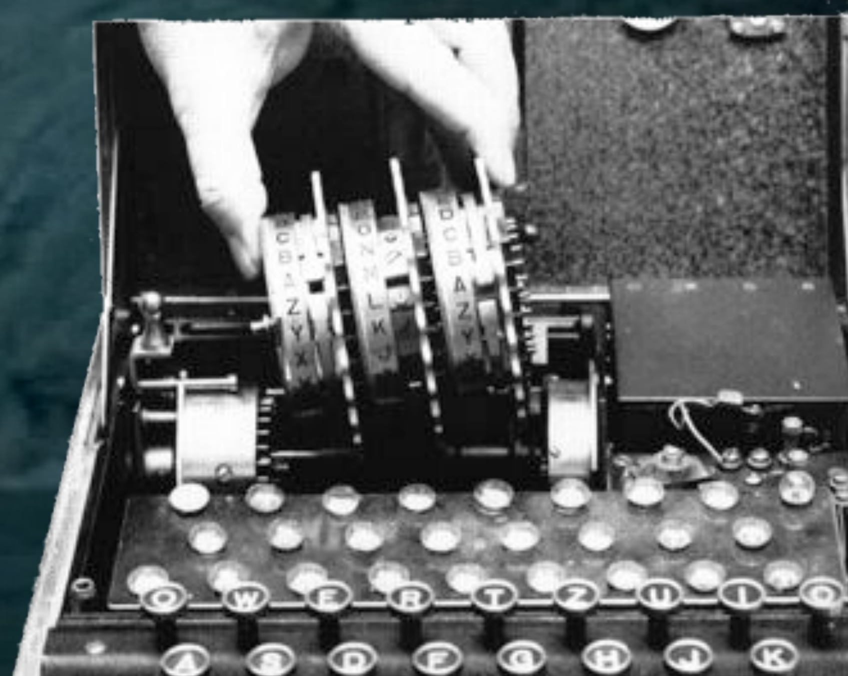


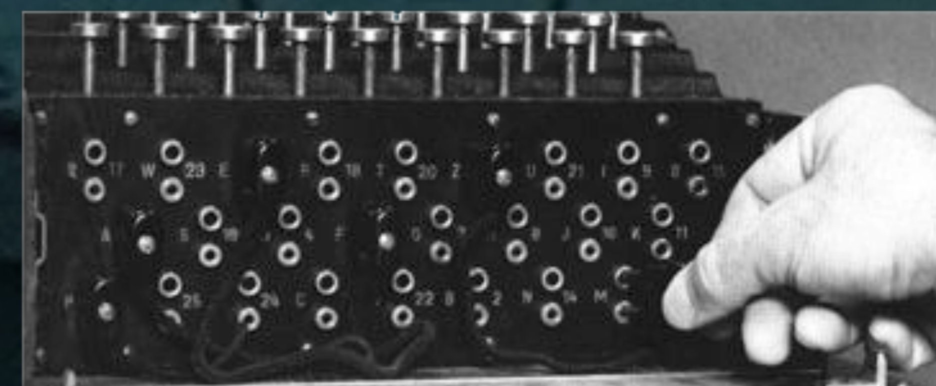

Important: if A → B then also B → A but never A → A

# Keying the Enigma







# Using the Enigma

Prüf=Nr. 4533

**Geheim!**

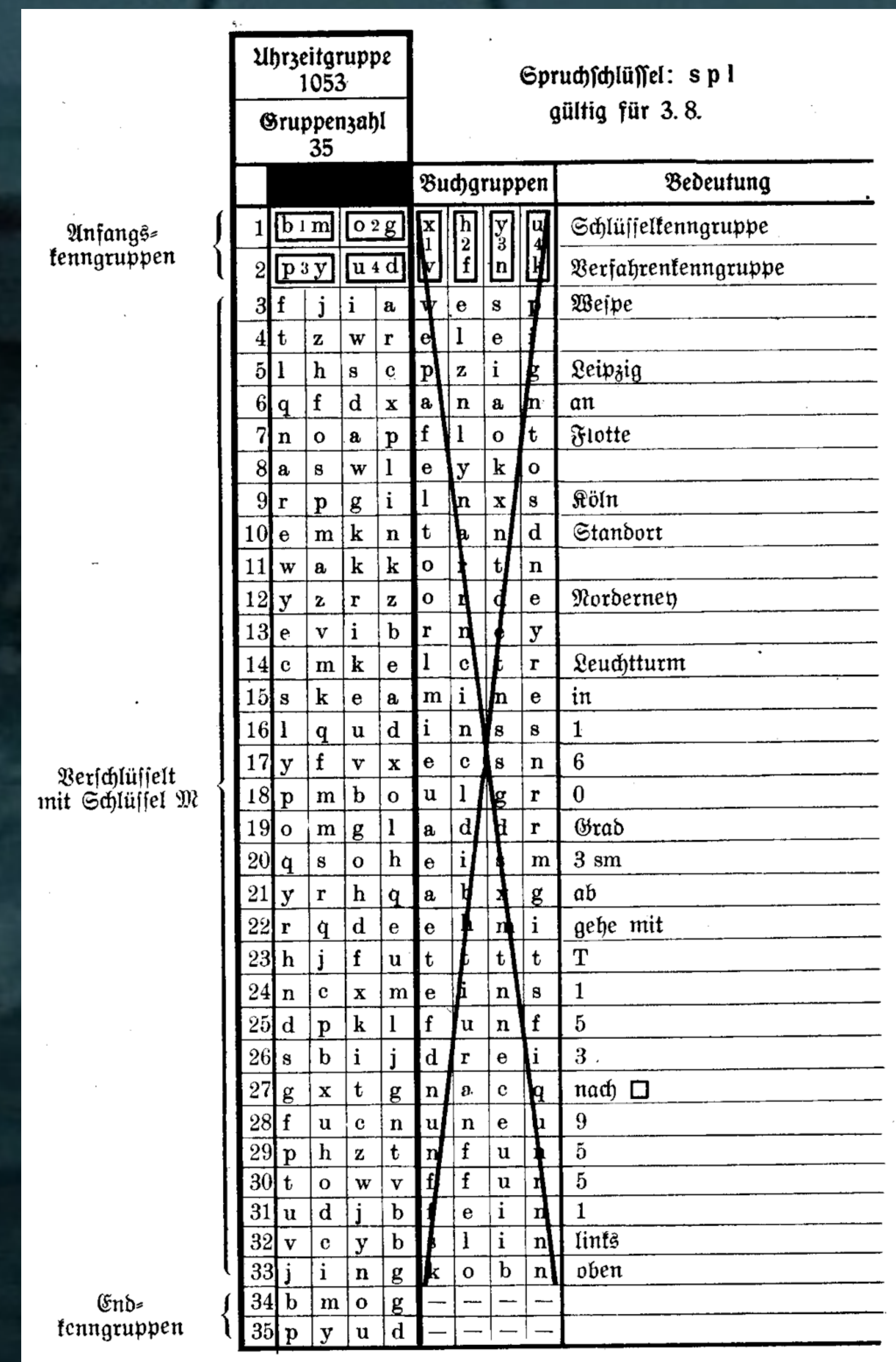
# Der Schlüssel M

## Verfahren M Allgemein

Oberkommando der Kriegsmarine
Berlin 1940

M. Dv. Nr. 32/1

Marineoberkommando Nordsee
Druckschriftenverwaltung



Uhrzeitgruppe 1053

Gruppenzahl 35

Spruchschlüssel: s p l
gültig für 3. 8.

| | | [redacted] | | | | Buchgruppen | | | | Bedeutung |
|---|---|---|---|---|---|---|---|---|---|---|
| Anfangs=kenngruppen | 1 | b 1 m | | o 2 g | | x 1 v | h 2 f | y 3 n | u 4 k | Schlüsselkenngruppe |
| | 2 | p 3 y | | u 4 d | | | | | | Verfahrenkenngruppe |
| Verschlüsselt mit Schlüssel M | 3 | f | j | i | a | w | e | s | p | Weſpe |
| | 4 | t | z | w | r | e | l | e | i | |
| | 5 | l | h | s | c | p | z | i | g | Leipzig |
| | 6 | q | f | d | x | a | n | a | n | an |
| | 7 | n | o | a | p | f | l | o | t | Flotte |
| | 8 | a | s | w | l | e | y | k | o | |
| | 9 | r | p | g | i | l | n | x | s | Köln |
| | 10 | e | m | k | n | t | a | n | d | Standort |
| | 11 | w | a | k | k | o | r | t | n | |
| | 12 | y | z | r | z | o | r | d | e | Norderney |
| | 13 | e | v | i | b | r | n | e | y | |
| | 14 | c | m | k | e | l | c | t | r | Leuchtturm |
| | 15 | s | k | e | a | m | i | n | e | in |
| | 16 | l | q | u | d | i | n | s | s | 1 |
| | 17 | y | f | v | x | e | c | s | n | 6 |
| | 18 | p | m | b | o | u | l | g | r | 0 |
| | 19 | o | m | g | l | a | d | d | r | Grad |
| | 20 | q | s | o | h | e | i | s | m | 3 sm |
| | 21 | y | r | h | q | a | b | x | g | ab |
| | 22 | r | q | d | e | e | h | m | i | gehe mit |
| | 23 | h | j | f | u | t | t | t | t | T |
| | 24 | n | c | x | m | e | i | n | s | 1 |
| | 25 | d | p | k | l | f | u | n | f | 5 |
| | 26 | s | b | i | j | d | r | e | i | 3 |
| | 27 | g | x | t | g | n | a | c | q | nach □ |
| | 28 | f | u | c | n | u | n | e | u | 9 |
| | 29 | p | h | z | t | n | f | u | n | 5 |
| | 30 | t | o | w | v | f | f | u | r | 5 |
| | 31 | u | d | j | b | f | e | i | n | 1 |
| | 32 | v | c | y | b | s | l | i | n | links |
| | 33 | j | i | n | g | k | o | b | n | oben |
| End=kenngruppen | 34 | b | m | o | g | — | — | — | — | |
| | 35 | p | y | u | d | — | — | — | — | |

# Enigma in action



Polish break

# The Asché documents

Gustave Bertrand

Hans Thilo Schmidt

# Rejewski breaks Enigma

- 1927/1928 first contact, analysis fails
- 1929 three mathematicians hired
- 1932 Rejewski breaks Enigma with help of the Asché documents
- bomby developed to mechanize solving
- 1 feb 1936 monthly rotor changes
  1 nov 1936 daily rotor changes
  1936-1938 number of networks grows
  1939 extra rotors added – solution halts
- 25-26 juli 1939 Pyry conference

Marian Rejewski 1905–1980

# How did he do it?

the message header was fatally flawed!

repaired after 1 may 1940 but then TOO LATE!!!

```
1755-135 WEP ULZNU HFIKLB SGEXU ...
1755   = Zeitgruppe
135    = number of letters from Kenngruppe
WEP    = Grundstellung
ULZNU  = Kenngruppe
HFIKLB = 2x enciphered Spruchschlüssel
```

Example: (1) set wheels to WEP then (2) choose Spruchschlüssel ABC (3) encipher ABCABC (4) result = HFIKLB then (5) set wheels to ABC

Rejewski could try for example ABC and see if it was correct and when deciphered reconstructed rotor wirings step by step

# Enigma substitution

- rotor effects monoalfabetic substitution
- A → P → R → F → A  is called cycle (APRF)
- complete alphabet in cycles (APRF)(GQZBJV)…()
- A → P en P → A is called involution cycle (AP)
- Enigma (AP)(ZI)(GE)..() is product of involutions
- 2x Enigma substitution results in paired cycles
  (AJUT)(KVZF)(Q)(M)…
- double encipherment Spruchschlüssel just this!

# Double encipherment

twice Spruchschlüssel ABC ABC → PQR XYZ

```
E1: A → P  and  P → A
E2: B → Q  and  Q → B
E3: C → R  and  R → C
E4: A → X  and  X → A
E5: B → Y  and  Y → B
E6: C → Z  and  Z → C
```

```
P→X is E4(A) = E4(E1(P)) or E4E1(P)→X
Q→Y is E5(B) = E5(E2(Q)) or E5E2(Q)→Y
R→Z is E6(C) = E6(E3(R)) or E6E3(R)→Z
```

three double encipherments E4E1, E5E2 and E6E3 !!!

# Completing cycles

- `??????` → `PQRXYZ` resulting in
  `E4E1 = (P,X,...)`
  `E5E2 = (Q,Y,...)`
  `E6E3 = (R,Z,...)`
- `??????` → `TYJGNZ` lengthens to
  `E5E2 = (Q,Y,N,...)` etc.
- many message headers →
  complete series of cycles

# Testing Spruchschlüssels

```
E4E1 = (DVPFKXGZYO)(EIJMUNQLHT)(BC)(RW)(A)(S)
E5E2 = (BLFQVEOUM)(HJPSWIZRN)(AXT)(CGY)(D)(K)
E6E3 = (ABVIKTJGFQNY)(DUZREHLXWPSMO)
```

question: could Spruchschlüssel be AAA?
answer: check if letters in opposite cycle of a pair

SUGSMF = AAAAAA? E5E2=UM not in (AXT)(CGY)

answer: NO

```
SYXSCW = AAAAAA? E4E1=SS in (A)(S)
                 E5E2=YC in (AXT)(CGY)
                 E6E3=XW in (ABV..)(DUZ..XW..)
                 answer: YES
```

# British break

15 minutes break - please be back on time

# Bletchley Park

# Hut codebrekers

# Alan Turing's bureau


zijn beertje

# Breaking Kriegsmarine Enigma

# Key events

1941 March: Lofoten raid gives first hint how to end BLACKOUT

1941 May: U110 captured codebooks arrive in Bletchley

1941 June: keys from München and Lauenburg exploited

1941 July: U-boat traffic broken

1942 February: Enigma M4 + codebook change ➔ BLACKOUT

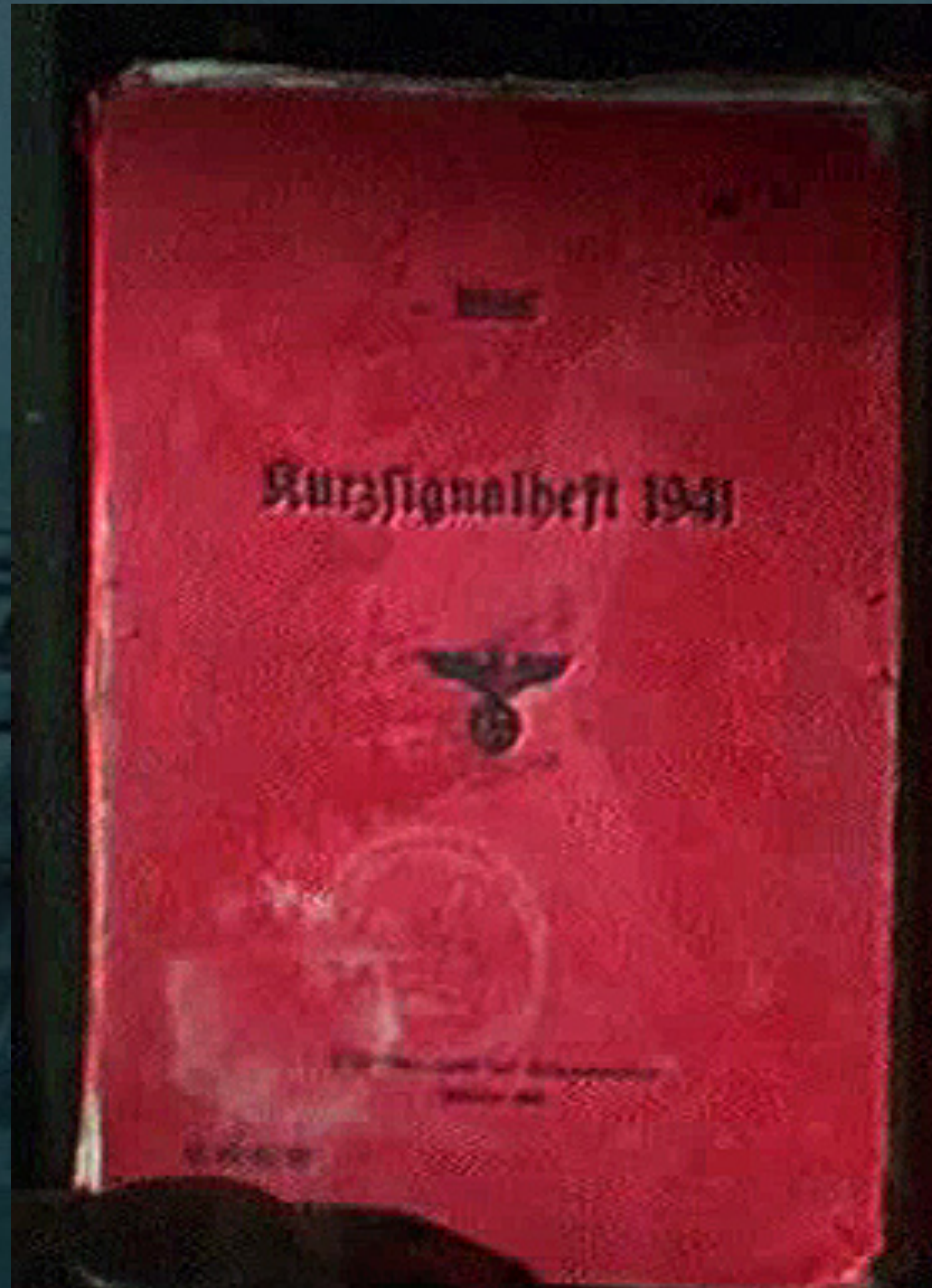
1942 November: Kurzsignalheft captured from U559

1942 December: U-boat traffic broken again

1943 March: Wetterkurzschlüssel changed ➔ BLACKOUT

1943 March: Convoy battle enables codebook reconstruction

1943 May: U-boats leave the North Atlantic




*Change of codebooks happened and required capture of the new one or reconstruction by cryptanalysis!*

# How they broke Enigma

## U-boats

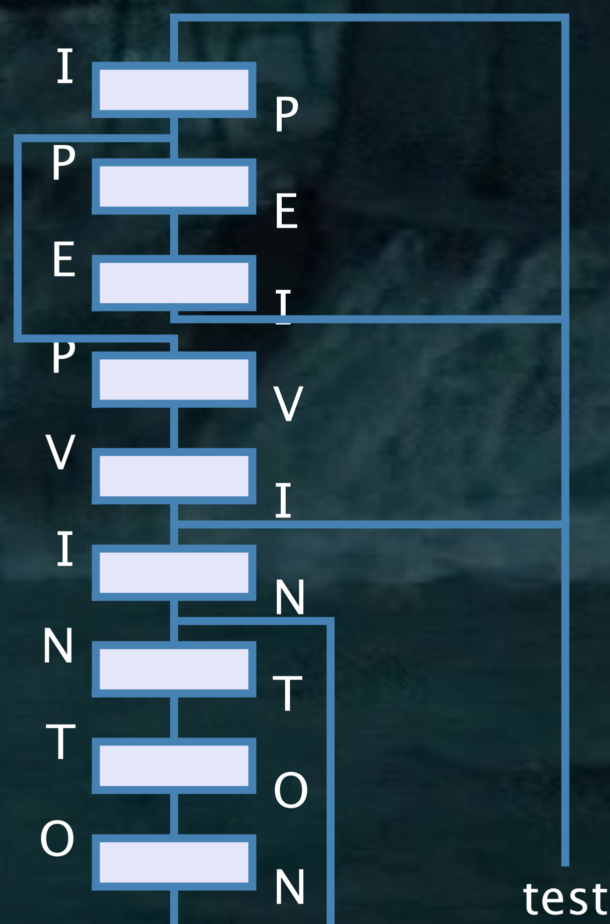
1. U-boat regularly observes weather conditions ➔ weather forecasts
2. Weather report is first encoded with Wetterkurzschlüssel codebook
3. Encoded weather report enciphered on Enigma with daily key
4. Enciphered report transmitted to U-boat headquarters

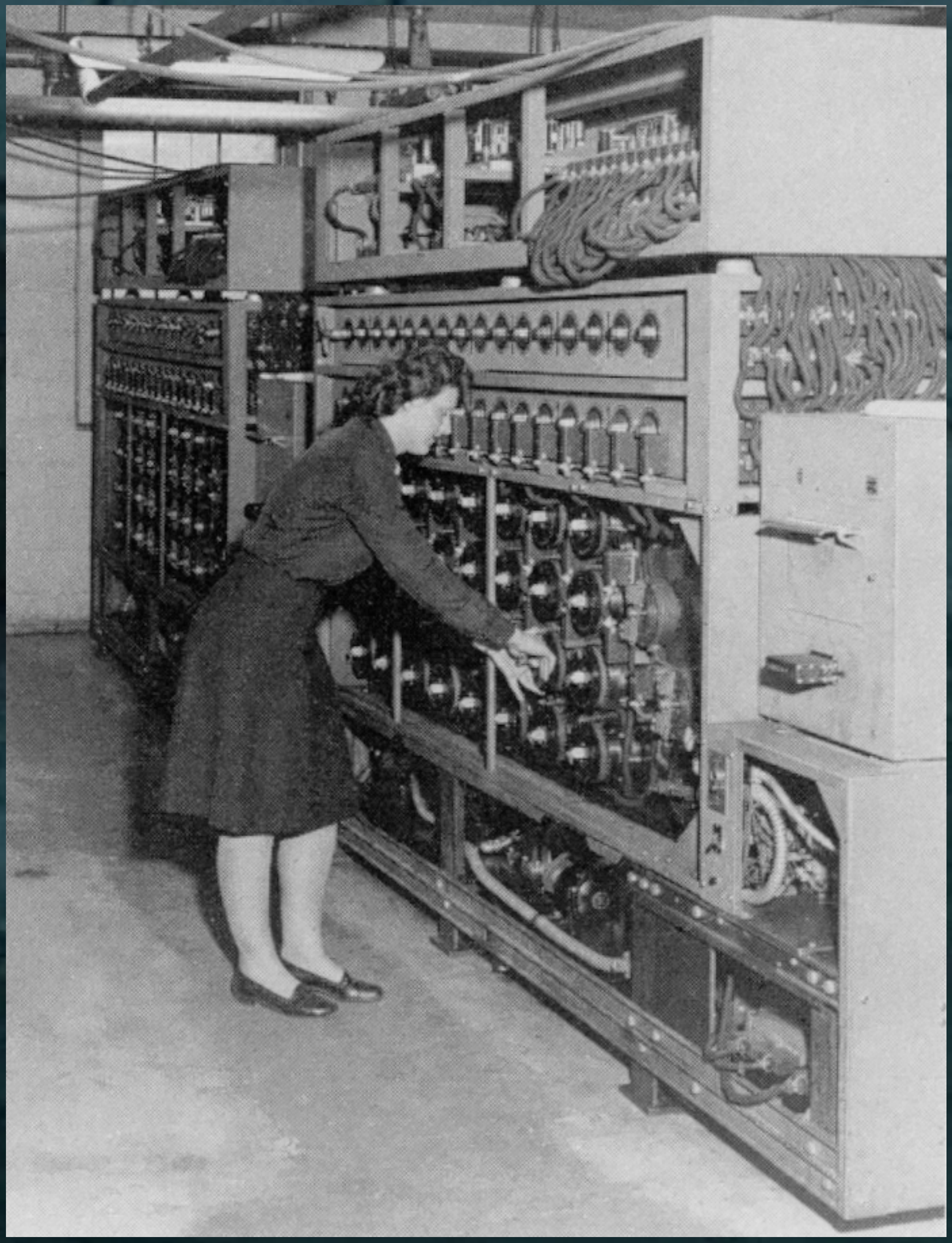
## Bletchley

5. U-boat position found by direction finding ➔ weather ➔ content = crib
6. Encode crib with captured Wetterkurzschlüssel
7. Try encoded crib on bombe machines ➔ Enigma key
8. Use this key to decrypt U-boat operational Enigma messages
9. Decode decrypted operational messages with captured Kurzsignalheft

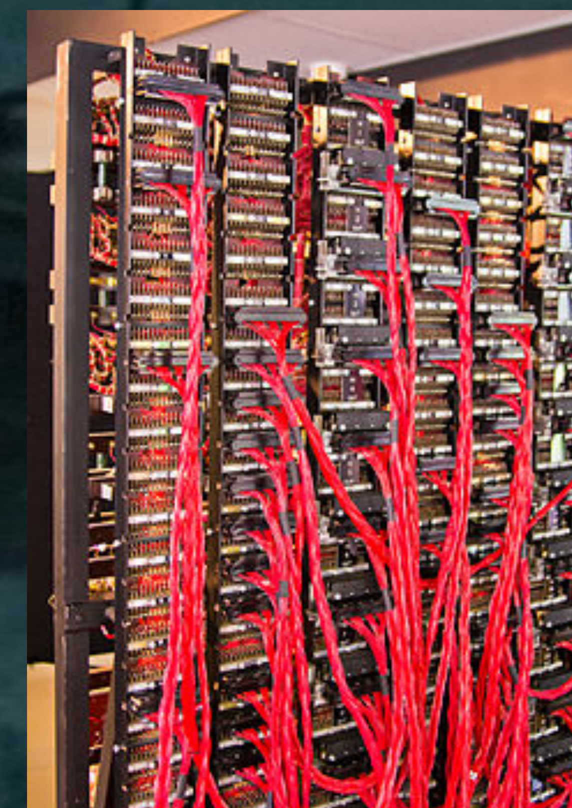
# Turing bombe tests crib

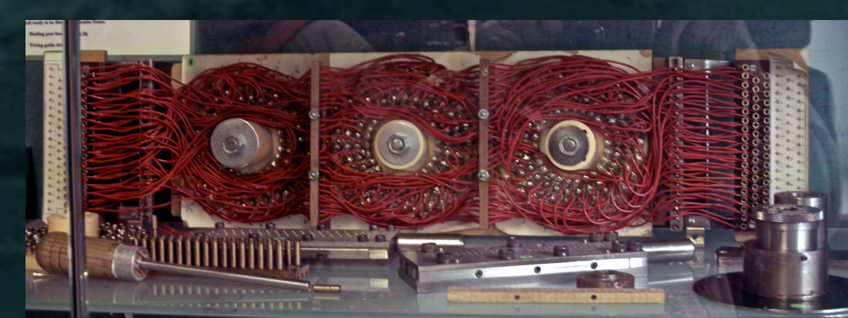
```
position:0  00  0  1    1  1         22
         3  56  8  0    3  5         12
enigma:..N  PV  I  P    I  T         OV
  crib:..ThEPrEsIdeNtOftheuNIt
            loops NTO,EPI
```

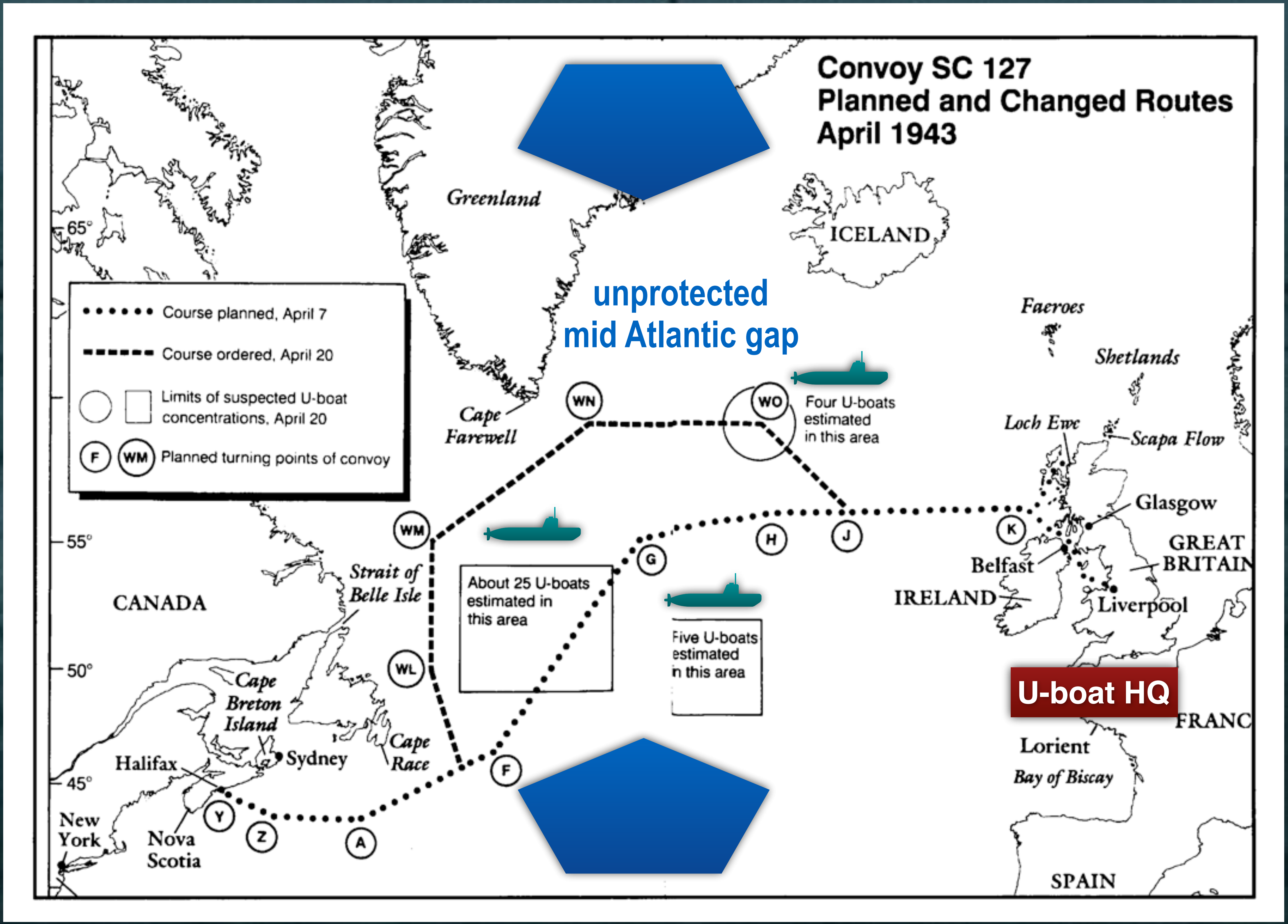
# Bombe rotors



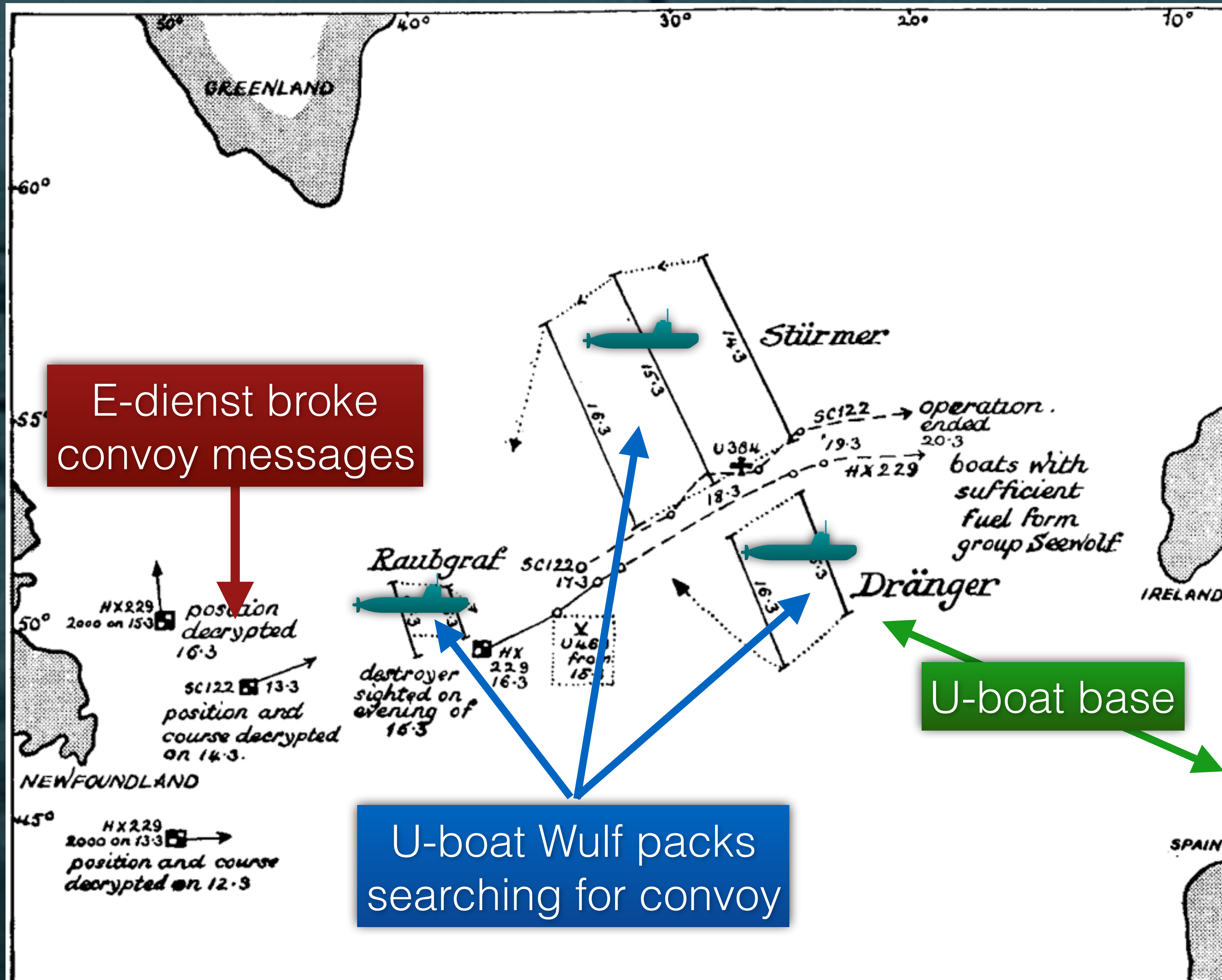



Attack and Defense

# Convoy routing



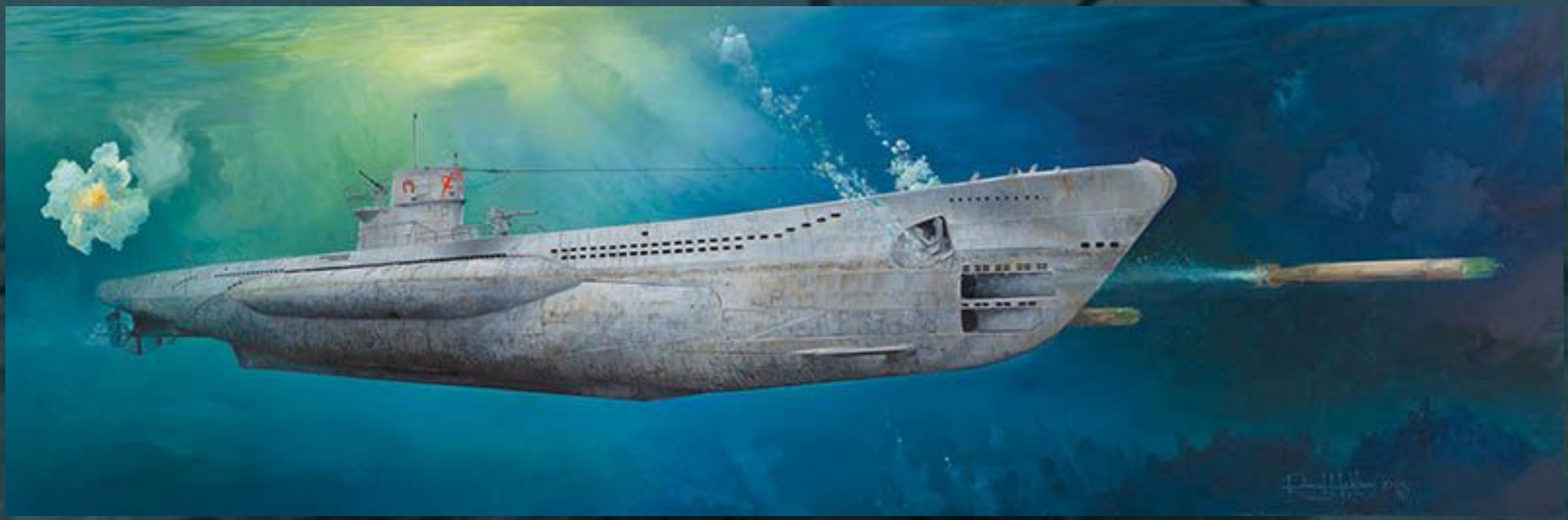
# U-boat tactics




Groups *Raubgraf*, *Stürmer* and *Dränger*, 15th–20th March, 1943.
Operations against SC 122 and HX 229

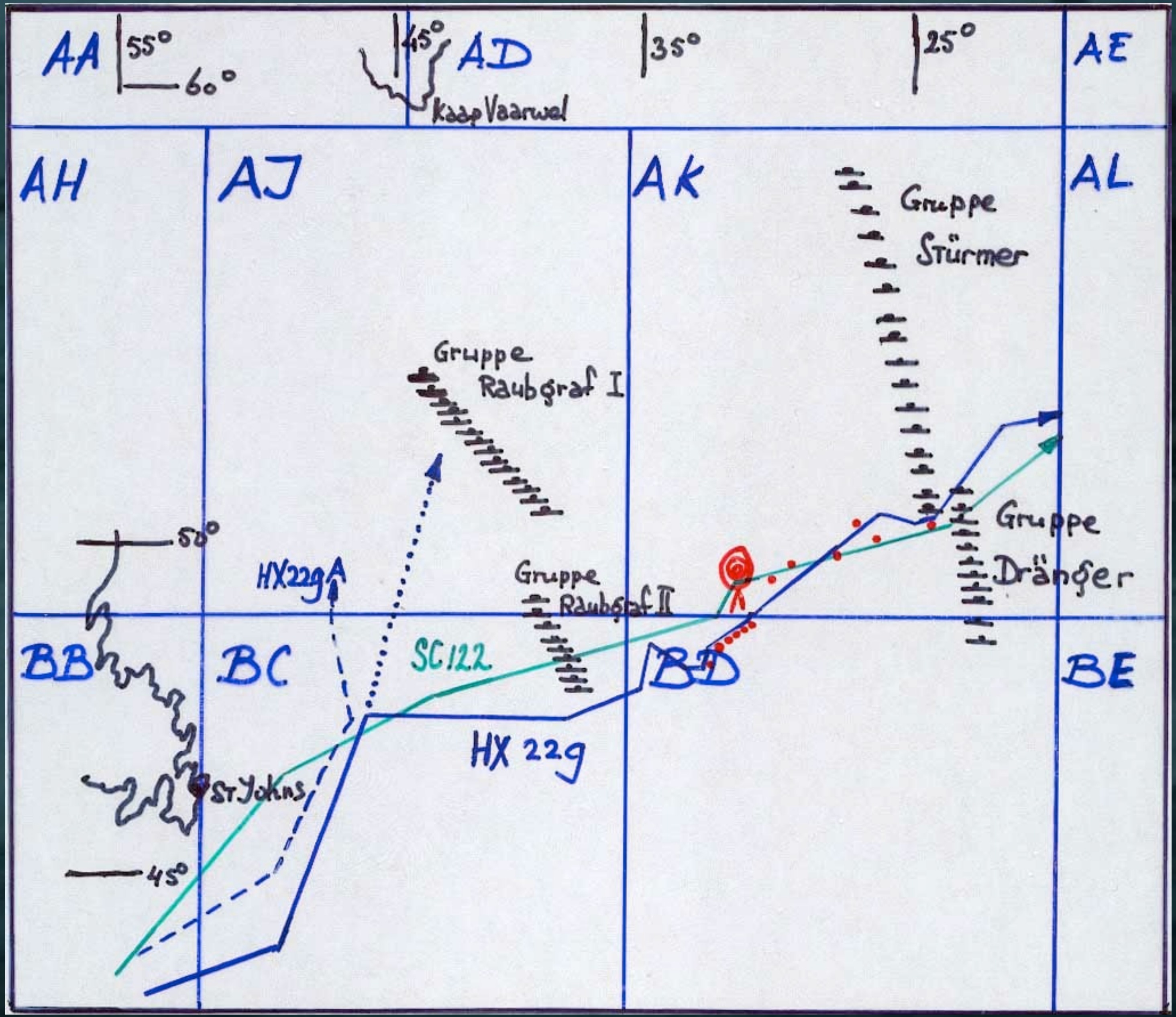

Discovery

# “Das Boot” U96 Type VIIC



- type VIIC Atlantic U-boat 1940-1944, built 567
- length 67,5 meter, crew 44
- speed surface 17,7 knots, range 9700-3450 miles  (1 knot = 1,850 km/hr)
- speed submerged 7,6 knots, range180-30 miles
- depth guaranteed 100 meter, more in practice
- 4 forward & 1 rear torpedo launch tubes, max 14 torpedo’s, 2 deck guns

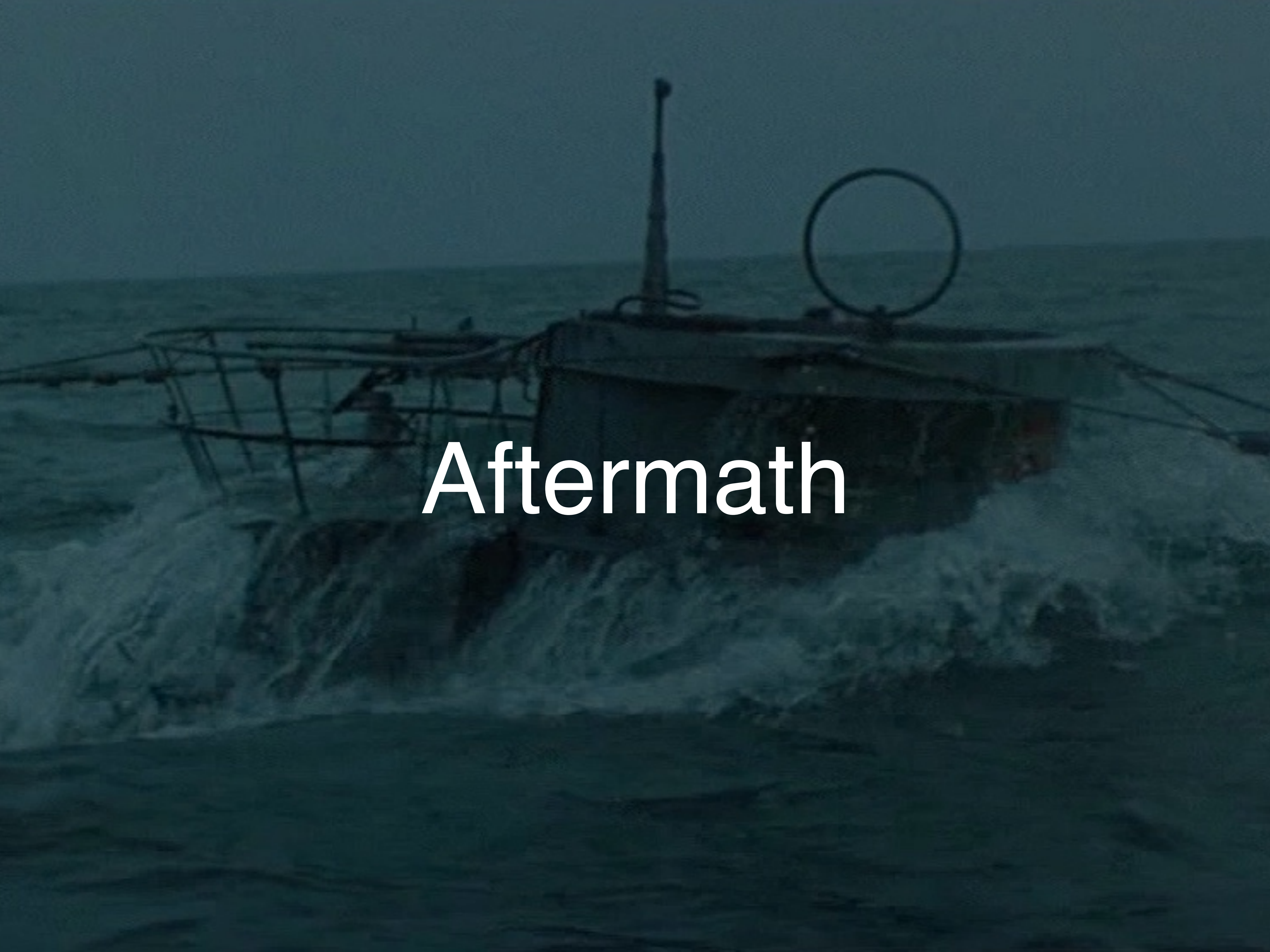
# Convoy battle march 1943



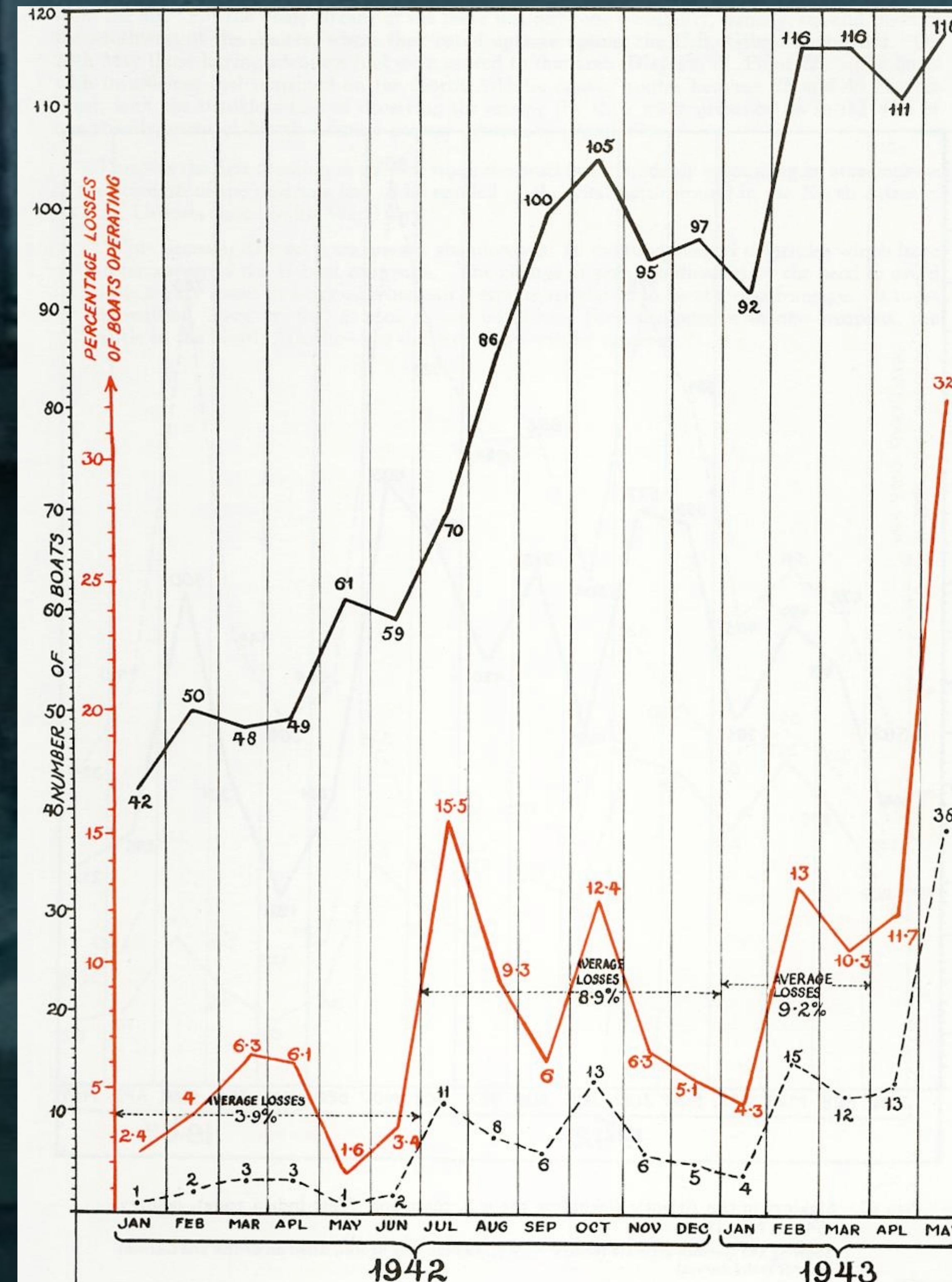

| signal U653<br>16 March 1943 07:25 | | |
|---|---|---|
| message | code | enigma |
| Feind im Sicht | CCHH | XGBT |
| BD1 | JNOS | GZZV |
| 491 | NBYU | QAGN |
| 70 Grad | QJRK | TBAB |
| Fahrt 10 sm | QRTU | ZVKL |

# What we will see

1. Emerging U-boat receiving “Akelei” code word
2. Tom (codebreaker) back at Bletchley - he is not welcome
3. Change of weathercode causes new blackout
4. Conference with navy staff and American advisor
5. Tom is blamed and will be ousted from Bletchley
6. Alarm sounds - U-boats to start a massive attack
7. Tom then finds a way to break Enigma again
8. Waiting for the signals to come in
9. Collect crucial data for decisive bombe runs


Aftermath

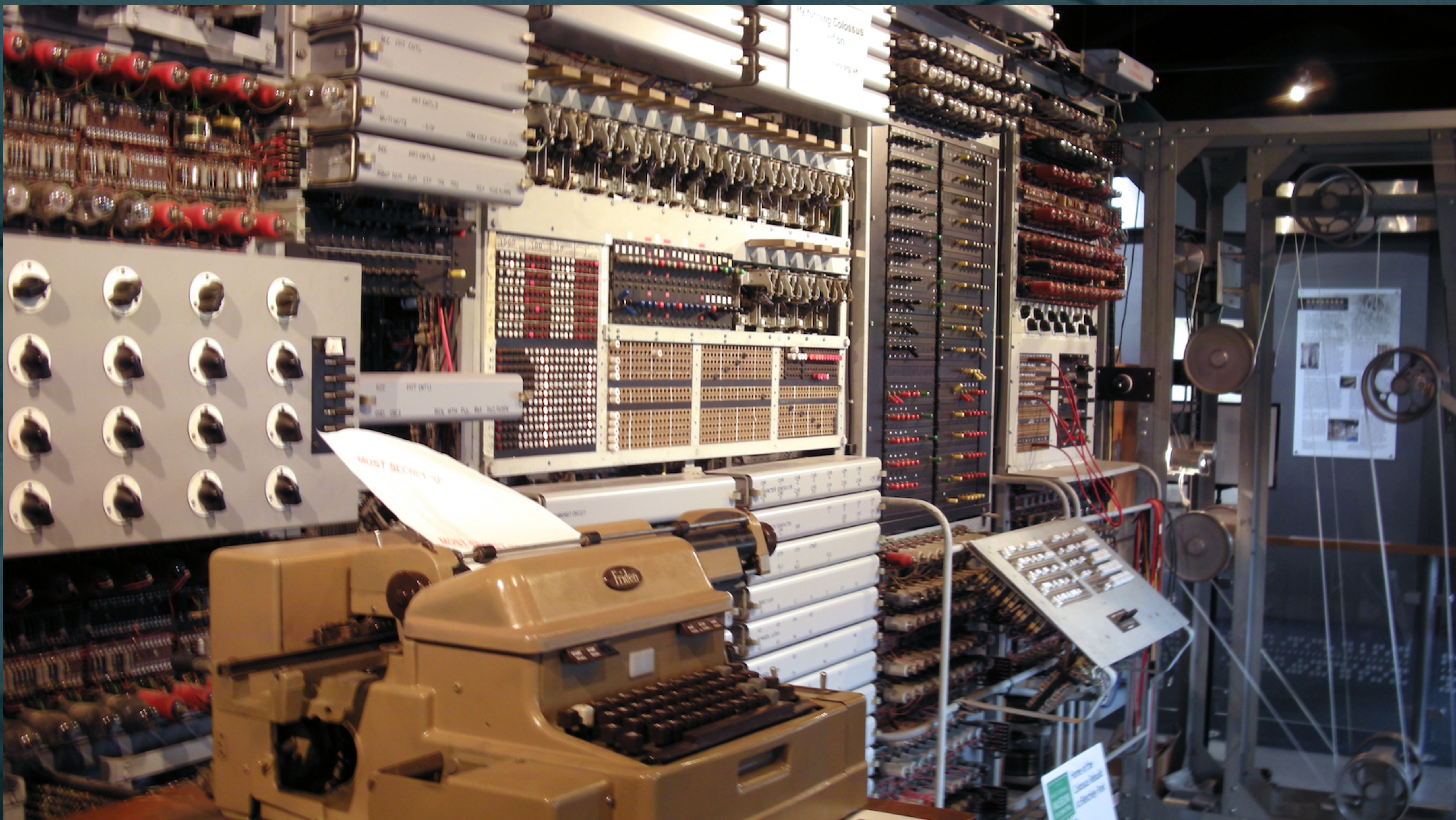
# Effect on U-boot losses




After May 1943 the German U-boats left the North Atlantic theatre

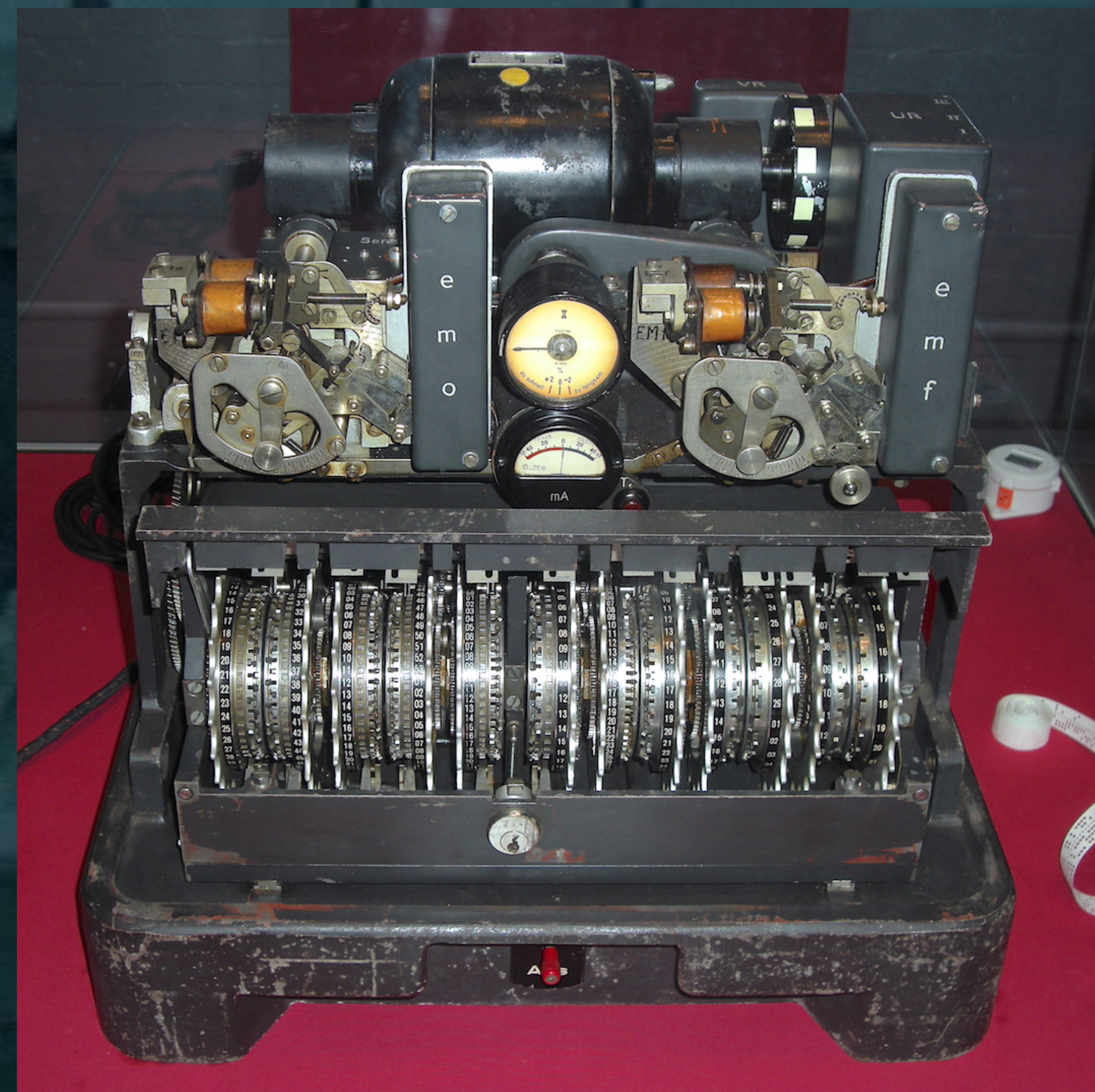
This enabled the buildup for the invasion in Normandy the following year

# Colossus

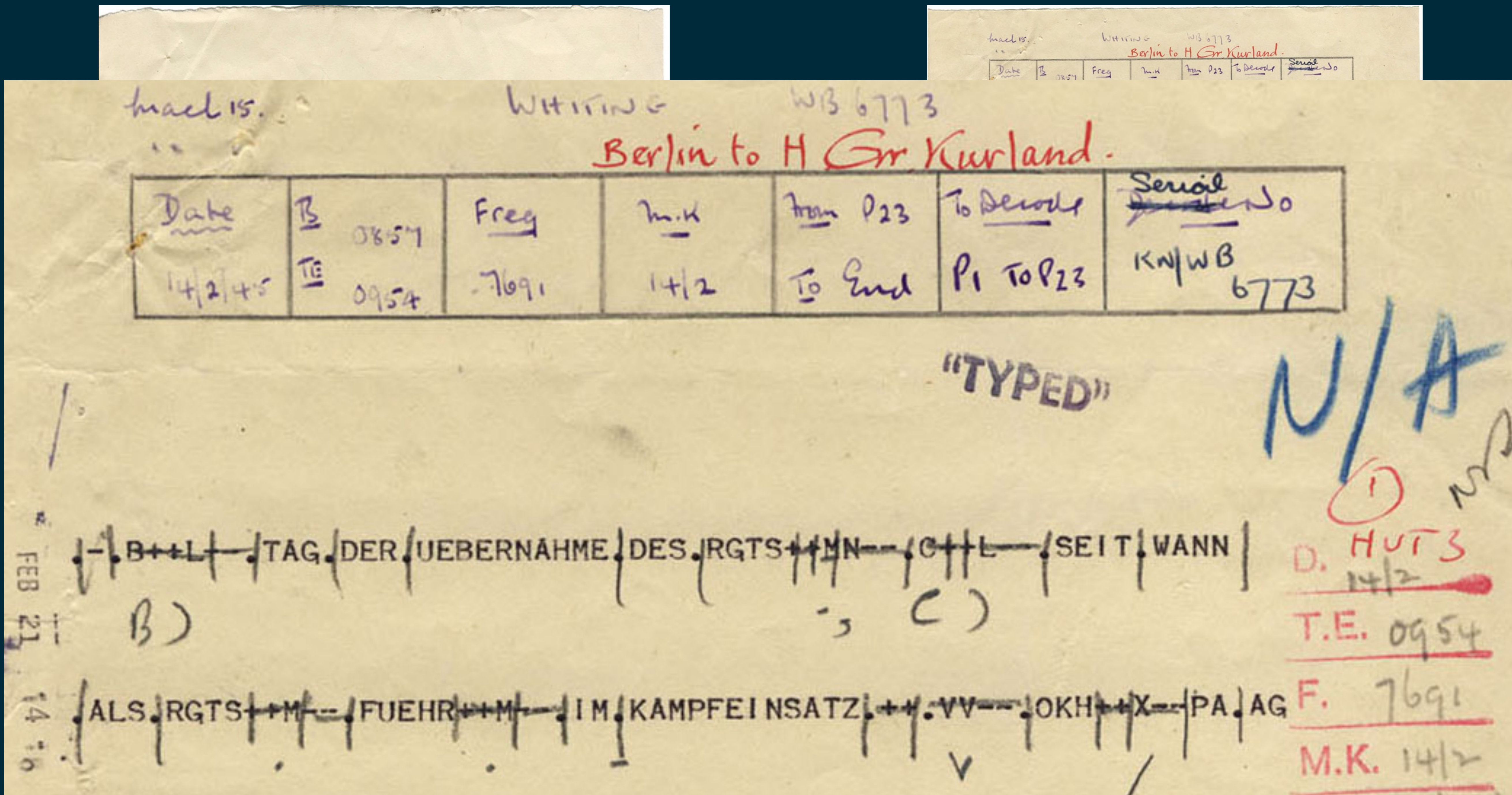


# Lorenz Schlüsselzusatz

Communications between
Hitler and his top
Army headquarters



# Bletchley Park




intercept sheet

decrypt

# Lessons in Security

- Human behaviour: Lazy German operators
- Blind eye: Machines breaking crypto machines
- Consistency: Weather and Signals code change
- Trust in numbers: Relying on many crypto keys
- Mathematics: Bad keying procedure

## Conclusion

in intelligence every aspect counts
each flaw can compromise everything

# Literature

- The U-Boat War in the Atlantic 1939-1945, HMSO Publications Centre, 1989
- Seekrieg im Äther - Die Leistungen der Marine-Funkaufklärung, Heinz Bonatz, 1981
- Die Wende im U-Boot Krieg - Ursachen und Folgen 1939-1943, Jochen Brennecke, 1984
- Intercept The Enigma War, Józef Garliński, 1979
- Geheimoperation Wicher, Władysław Kozaczuk, Jürgen Rohwer, 1989
- The Hut Six Story - Breaking the Enigma Codes, Gordon Welchman, 1982
- Code Breakers - The Inside Story of Bletchley Park, F.H. Hinsley and Alan Stripp, 1993
- Seizing the Enigma - The Race to Break the German U-Boat Codes, David Kahn, 1991
- Cryptography college notes url = www.hvandermeer.com

# Classical Cryptography

## Stream ciphers

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.2, 2020/02/25 12:58:59 UTC)

Thursday, February 27, 2020

1. Running keys
2. One time pads
3. Autokeys
4. Modes of operation

# Outline

1 Running keys

2 One time pads

3 Autokeys

4 Modes of operation

# Keytexts and running keys

- Progressive and keyword based polyalphabetic ciphers are still susceptible to certain statistical attacks after some preprocessing
- Alternative is to use a **keytext**
  - Can be very long when produced from a book
  - Now standard statistics fails because the key material does not repeat
  - We call this **running key** ciphers
  - Also called a **keystream** in a more general setting
    - when it need not be derived from a normal text

# Cryptanalysis for running keys

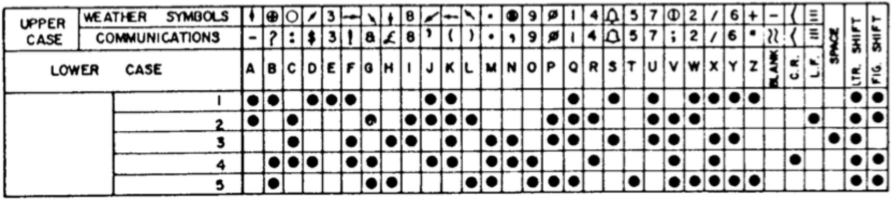
- If you have **lots** of ciphertext from **different** messages (same key)
  - Use the **kappa**($\kappa$)-test to check for **superimposition**
  - Apply the **phi**($\varphi$)-test to see whether columns are **mono**alphabetic
  - Apply the **chi**($\chi$)-test to see whether columns can be **combined**
- If you have a **single** ciphertext encrypted using a **keytext**
  - Use statistics on pairs of letters in the same position

    in keytext and plaintext
  - Both of these letters are based on the language letter distribution
- Use the probable word or **crib** method
  - Can be applied to both plaintext and keytext

# Outline


1. Running keys
2. One time pads
3. Autokeys
4. Modes of operation

# One-time pad (1)




- The ultimate **random** running key
  - Named **one-time pad** or **Vernam** cipher
  - Or one-time tape in case of (**Baudot** coded) paper tapes
  - But it is only safe when it is **never reused**

# One-time pad (2)

- We obtain **perfect security** when used properly
  - This is security in the **information theoretic** sense
- The key material is at least as long as the message
  - So it cannot be used easily for bulk encryption
  - Use it only for short and very important messages

# Outline


1. Running keys
2. One time pads
3. Autokeys
4. Modes of operation

# Autokey: between repeating and running keys

- Cardano's **autokey** cipher
- Uses the plaintext as the key material
- First plaintext word is encrypted with itself as key
- Later plaintext words are encrypted by using the plaintext again from the beginning as the key
- Cardano's system has some problems
  - It is a keyless system
  - There is ambiguity in decoding
  - There might be synchronisation problems when decoding
    - After an early decoding mistake we get **error propagation**

# Cardano's autokey example

```
plaintext:  cardano autokey is a weak cipher undoubtedly
keystream:  cardano cardano ca c card cardan cardanoauto
ciphertext: EAIGAAC CUKRKRM KS C YERN EIGKEE WNURUOHEXEM
```

Figure 1: Cardano's system is a "Per word Vigenère"-cipher

In modern encoding. Holden's book uses legacy encoding.

# Vigenère's improvements (1)

- Use a "priming key" or **initialization vector (IV)**[1]
  - Now the system is no longer keyless or ambiguous
- Also add a "method of alteration" to produce a better key
  - Use transformed plaintext letters in the key letter sequence
- The system is called a **plaintext autokey** cipher
- Note that it does not solve the error propagation issue

[1] Maybe **seed** is a better terminology, since the IV is secret in this case

# Plaintext autokey example

```
plaintext:  vigenerehasthoughtofanimprovement
keystream:  ivvigenerehasthoughtofanimproveme
ciphertext: DDBMTIEIYEZTZHBUBZVYOSIZXDDMSHIZX
```

Figure 2: Using a seed or secret initialization vector ("iv" in this case)

```
plaintext:  vigenerehasthoughtofanimprovement
ivshifted:  ivvigenerehasthoughtofanimproveme
keystream:  reertvmvivszhgslftsgluzmrnkilevnv
ciphertext: MMKVGZDZPVKSOUMRMMGLLHHYGEYDPQZAO
```

Figure 3: Plaintext processed using atbash for key alteration

# Vigenère's improvements (2)

- Better synchronization by using a **ciphertext autokey** cipher
- This solves the error propagation issue
- A big problem is that the keystream is in view all the time
    - With a little trial and error the system can be cracked
    - So it is even more important to add a "method of alteration" as a key

# Ciphertext autokey example

```
plaintext:  vigenerehasthoughtofanimprovement
keystream:  seednmkhaqblhqteoenkvxbpvkjbkbxwo
ciphertext: NMKHAQBLHQTEOENKVXBPVKJBKBXWONBJH
```

Figure 4: Using “seed” as seed before using the ciphertext as key

```
plaintext:  vigenerehasthoughtofanimprovement
seeded:     seedcdbakapdwzcpkorqwewodilxlicxs
keystream:  hvvwxwyzpzkwdaxkplijdvdlwrocorxch
ciphertext: CDBAKAPDWZCPKORQWEWODILXLICXSDBPA
```

Figure 5: Ciphertext processed using atbash for key alteration

# Key autokey ciphers

- **Seed** the keystream with a block of letters
- Transform the current key block into the next key block

  by using some simple enciphering
- Drawback is that this system will still be (ultimately) periodic
- But the period can be quite long with respect to the seed length

# Key autokey example

```
plaintext:  autokeyingfromthekeywhynot
keystream:  startswdfgszgttscjhgsfmvts
ciphertext: SNTFDWULSMXQUFMZGTLEOMKIHL
```

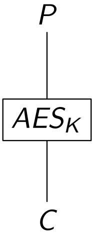
Figure 6: Using “start” as seed with additive “addon”

Can you find the period of this key?

# Outline


1. Running keys
2. One time pads
3. Autokeys
4. Modes of operation

# Block ciphers




For instance the block cipher AES transforms

some block of bits into a similar block of bits

It depends on a strong secret key K

# Block cipher modes

- **ECB**: Electronic CodeBook mode
  - A monoalphabetic substitution on an alphabet of blocks
- **CFB** (**CBC**): Cipher FeedBack (Block Chaining) mode
  - Similar to a **ciphertext autokey** system
- **PFB** (**PBC**): Plaintext FeedBack (Block Chaining[2]) mode
  - Similar to a **plaintext autokey** system
- **OFB**: Output FeedBack mode
  - Similar to a **key autokey** system
- **CTR**: Counter mode
  - A modern keystream generator

[2] The Block Chaining modes are variants of the FeedBack modes

# ECB mode

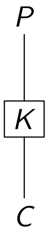



Figure 7: ECB: encryption of plaintext block is always the same

This system is monoalphabetic with a (very) large alphabet

# CFB mode building block

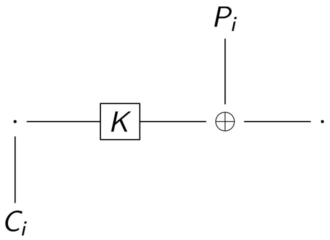



Figure 8: Ciphertext autokey based system

In our example we had K is atbash and $\oplus$ is Vigenère

# CFB mode encryption

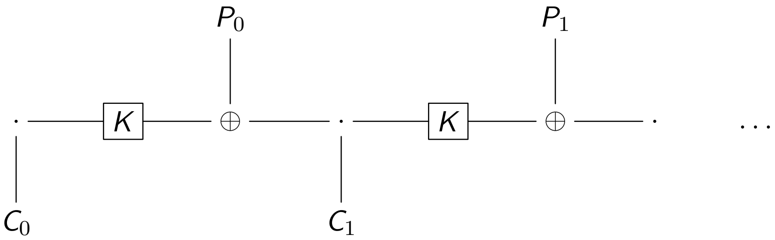



Figure 9: CFB: $C_0$ is the initialization vector

In general K is a key for some block cipher and $\oplus$ is xor

And the IV does not need to be secret

# CBC mode building block

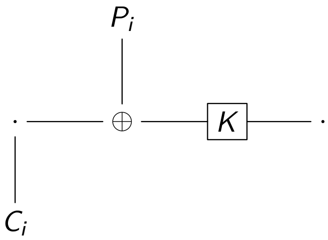



Figure 10: Ciphertext autokey *variant*

Can you see how?

# CBC mode encryption

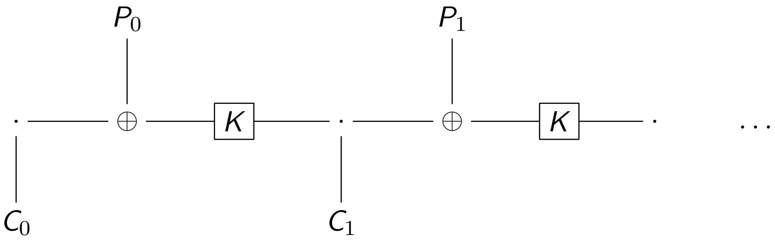



Figure 11: CBC: $C_0$ is the initialization vector

In general K is a key for some block cipher and $\oplus$ is xor

And the IV does not need to be secret

# PFB mode building block

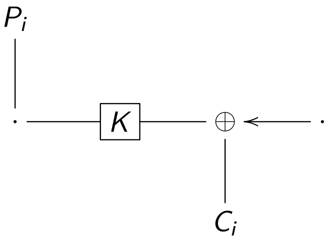



Figure 12: Plaintext autokey based system

In our example we had K is atbash and $\oplus$ is Vigenère

# PFB mode encryption

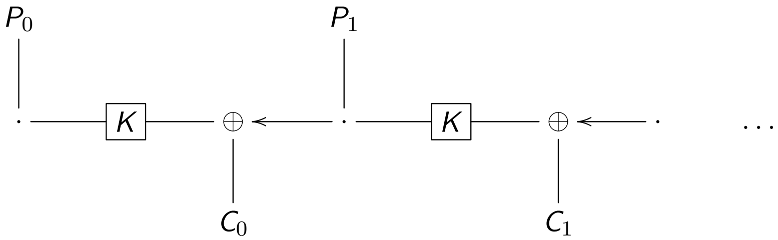



Figure 13: PFB: $P_0$ might be a seed or initialization vector

In our example we had K is atbash and $\oplus$ is Vigenère

# PBC mode building block

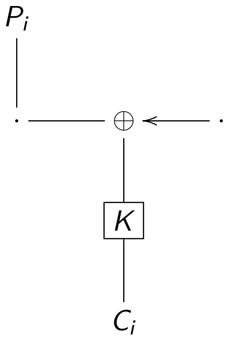



Figure 14: Plaintext autokey *variant*

# PBC mode encryption

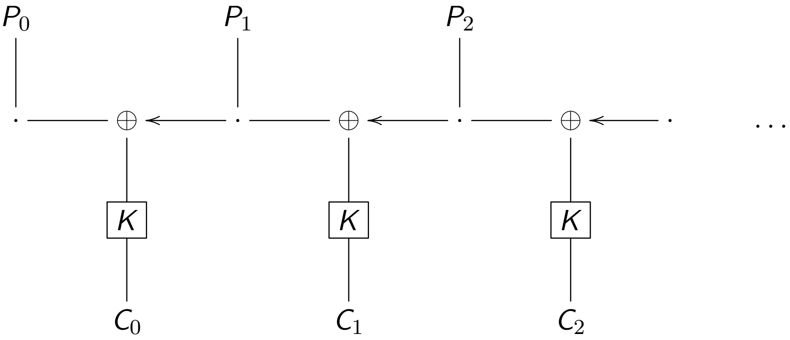



Figure 15: PBC: $P_0$ might be a seed or initialization vector

# PFB-variant mode building block

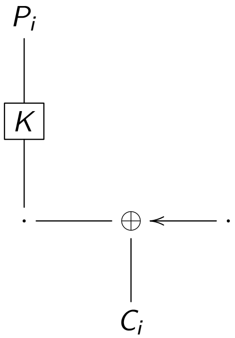



Figure 16: Yet another plaintext autokey *variant*

The difference with normal PFB is rather subtle. See also next slide.

# PFB-variant mode encryption

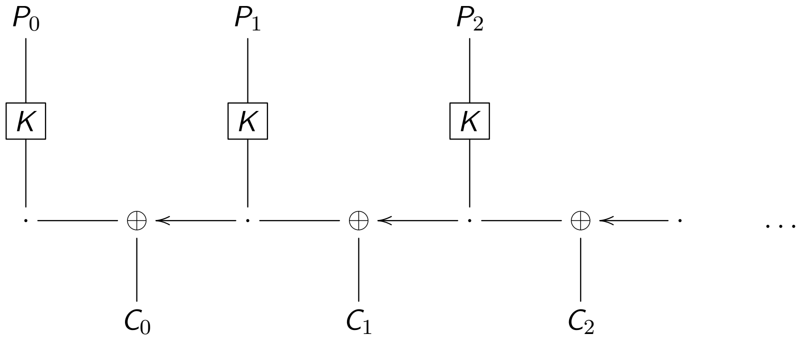



Figure 17: PFB-variant: $P_0$ might be a seed or initialization vector

# OFB mode building block

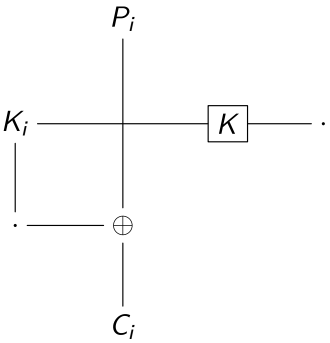



Figure 18: OFB: Key autokey based

# OFB mode encryption

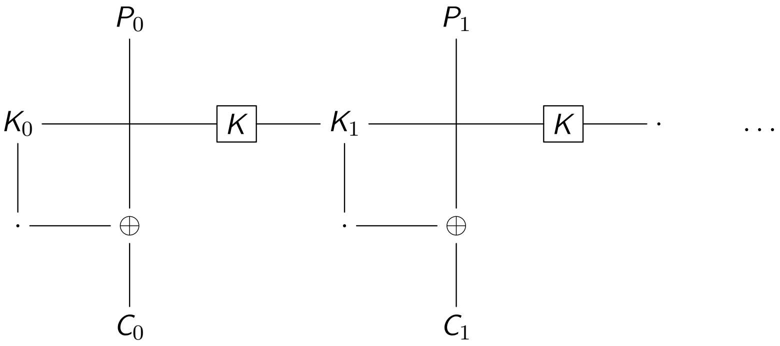



Figure 19: OFB: $K_0$ is a key seed

# CTR mode building block

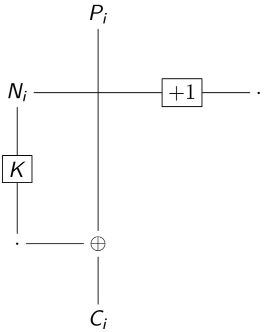



Figure 20: CTR: Encryption of a simple counter

# CTR mode encryption

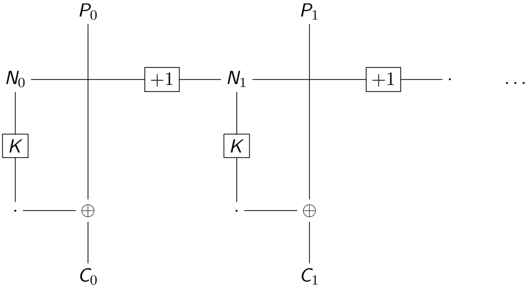



Figure 21: CTR: Keystream with counter initialization $N_0$

Note that this is easily parallellizable

# Classical Cryptography

## Basic number theory

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.1, 2020/02/25 13:18:32)

Monday, March 2, 2020

# Table of Contents


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Outline

1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# The field $\mathbb{Q}$ of rational numbers (1)

$$\mathbb{Q} = \{\frac{p}{q} | p \in \mathbb{Z}, q \in \mathbb{N}_{>0}, \text{gcd}(p,q) = 1\}$$

On $\mathbb{Q}$ we can define addition $+$ with neutral element $0$.

## Laws

$$\forall x \forall y (x + y = y + x) \quad \text{(Commutativity)}$$

$$\forall x \forall y \forall z ((x + y) + z = x + (y + z)) \quad \text{(Associativity)}$$

$$\forall x (x + 0 = x) \quad \text{(Neutral element)}$$

$$\forall x \exists y (x + y = 0) \quad \text{(Existence of inverses)}$$

# The field $\mathbb{Q}$ of rational numbers (2)

$$\mathbb{Q} = \{\frac{p}{q} | p \in \mathbb{Z}, q \in \mathbb{N}_{>0}, \gcd(p, q) = 1\}$$

On $\mathbb{Q}$ we can also define multiplication $\cdot$ with neutral element $1$.

## Laws

$$\forall x \forall y (x \cdot y = y \cdot x) \quad \text{(Commutativity)}$$

$$\forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z)) \quad \text{(Associativity)}$$

$$\forall x (x \cdot 1 = x) \quad \text{(Neutral element)}$$

$$\forall x \neq 0 \exists y (x \cdot y = 1) \quad \text{(Existence of inverses)}$$

# The field $\mathbb{Q}$ of rational numbers (3)

## Law

$$\forall x \forall y \forall z ((x + y) \cdot z = (x \cdot z) + (y \cdot z)) \qquad \text{(Distributivity)}$$

## Non-law (wrong)

$$\forall x \forall y \forall z ((x \cdot y) + z = (x + z) \cdot (y + z)) \qquad \text{(Wrong way distributivity)}$$

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Primes and unique factorisation

$$\begin{aligned} \mathbb{P} &= \{2, 3, 5, 7, 11, 13, \ldots\} \\ &= \{p_0, p_1, p_2, p_3, p_4, p_5, \ldots\} \end{aligned}$$

**Theorem**

*Every natural number $n > 0$ can be written in an essentially unique way as a product of primes:*

$$n = \prod_{i=0}^{k-1} p_i^{a_i}$$

*where $a_{k-1} > 0$ if $k > 0$*

# Example prime factorisations

**Example**

$$210 = 2 \cdot 3 \cdot 5 \cdot 7 = 2^1 \cdot 3^1 \cdot 5^1 \cdot 7^1 = p_0^1 \cdot p_1^1 \cdot p_2^1 \cdot p_3^1$$

**Example**

$$189 = 3 \cdot 3 \cdot 3 \cdot 7 = 2^0 \cdot 3^3 \cdot 5^0 \cdot 7^1 = p_0^0 \cdot p_1^3 \cdot p_2^0 \cdot p_3^1$$

These factorizations give an isomorphism between $\mathbb{N}_{>0}$ and $\bigoplus_\omega \mathbb{N}$

where

$$\bigoplus_\omega \mathbb{N} = \{< a_0, a_1, \ldots > \mid \text{only finitely many } a_i \in \mathbb{N} \text{ are not zero}\}$$

# Outline

1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Multiplication table examples

## for $\mathbb{Z}_n \backslash \{0\}$

**Example ($\mathbb{Z}_6 \backslash \{0\}$)**

| $\cdot$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 5 |
| 2 | 2 | 4 | 0 | 2 | 4 |
| 3 | 3 | 0 | 3 | 0 | 3 |
| 4 | 4 | 2 | 0 | 4 | 2 |
| 5 | 5 | 4 | 3 | 2 | 1 |

**Example ($\mathbb{Z}_5 \backslash \{0\}$)**

| $\cdot$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 4 | 1 | 3 |
| 3 | 3 | 1 | 4 | 2 |
| 4 | 4 | 3 | 2 | 1 |

Working modulo 6 gives nasty zero divisors (resulting values in $\mathbb{Z}_6 = (\mathbb{Z}_6 \backslash \{0\}) \cup \{0\}$), but working modulo 5 (a prime) seems to behave much better (results in $\mathbb{Z}_5 \backslash \{0\}$).

# Prime fields

## Theorem

$\mathbb{F}_p = <\mathbb{Z}_p, +, \cdot, 0, 1>$ *is a field if and only if* $p$ *is prime.*

$\mathbb{Z}_n^* = <\{a \in \mathbb{Z}_n \mid gcd(a, n) = 1\}, \cdot, 1>$ *is a group for all* $n \in \mathbb{N}, n > 1$.

## Example ($\mathbb{Z}_{10}^*$)

| $\cdot$ | 1 | 3 | 7 | 9 |
|---|---|---|---|---|
| 1 | 1 | 3 | 7 | 9 |
| 3 | 3 | 9 | 1 | 7 |
| 7 | 7 | 1 | 9 | 3 |
| 9 | 9 | 7 | 3 | 1 |

## Example ($\mathbb{Z}_{12}^*$)

| $\cdot$ | 1 | 5 | 7 | 11 |
|---|---|---|---|---|
| 1 | 1 | 5 | 7 | 11 |
| 5 | 5 | 1 | 11 | 7 |
| 7 | 7 | 11 | 1 | 5 |
| 11 | 11 | 7 | 5 | 1 |

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Euler's $\varphi$-function and the Euler-Fermat theorem

## Definition ($n \in \mathbb{N}, n > 1$)

$\varphi(n)$ is the number of elements of $\mathbb{Z}_n^*$:

$$\varphi(n) = |\mathbb{Z}_n^*|$$

## Theorem

*For all $a \in \mathbb{Z}_n^*$ (or in other words gcd(a, n) = 1)*

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

## Example

$$\varphi(10) = 4$$

$$\varphi(12) = 4$$

# More properties of Euler's $\varphi$-function

## Theorem

$\varphi(p) = p - 1$, *for all primes p*

$\varphi(p^k) = p^{k-1}(p - 1)$, *for all primes p and* $k > 0$

$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$,

*for all relatively prime m and n*

## Corollary

*If p and q are different primes and* $N = pq$, *then* $\varphi(N) = (p - 1)(q - 1)$

In general, $\varphi(N)$ is easy to calculate if you know the factorisation of $N$.

# Cyclicity properties of $\mathbb{Z}_n^*$

**Example ($\mathbb{Z}_8^*$ is not cyclic)**

$$3^1 = 3; 3^2 = 1$$

$$5^1 = 5; 5^2 = 1$$

$$7^1 = 7; 7^2 = 1$$

All elements except 1 have order 2.

**Theorem**

*$\mathbb{Z}_p^*$ is cyclic of order $p - 1$ for all primes $p$.*

*We have an isomorphism for every prime $p$ (after choosing a generator $g$)*

$$< \mathbb{Z}_{p-1}, +, 0 > \cong < \mathbb{Z}_p^*, \cdot, 1 >: x \mapsto g^x$$

Warning: This isomorphism is easy to calculate from left to right but hard from right to left!

# Multiplicative order and primitive roots

**Example ($\mathbb{Z}_7^*$ is cyclic of order 6)**

$$2^0 = 1; 2^1 = 2; 2^2 = 4; 2^3 = 1$$

$$3^0 = 1; 3^1 = 3; 3^2 = 2; 3^3 = 6; 3^4 = 4; 3^5 = 5; 3^6 = 1$$

$$4^0 = 1; 4^1 = 4; 4^2 = 2; 4^3 = 1$$

$$5^0 = 1; 5^1 = 5; 5^2 = 4; 5^3 = 6; 5^4 = 2; 5^5 = 3; 5^6 = 1$$

$$6^0 = 1; 6^1 = 6; 6^2 = 1$$

3 and 5 are **primitive roots** or **generators**, having maximal order 6

2 and 4 have order 3, while 6 has order 2

$$< \mathbb{Z}_6, +, 0 > \cong < \mathbb{Z}_7^*, \cdot, 1 >: x \mapsto 3^x$$

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Pohlig-Hellman cipher

- Let $p$ be a prime and $e < p$ such that $\gcd(e, p-1) = 1$
- We encrypt numbers in $\mathbb{Z}_p^*$ (represented by $1, \ldots, p-1$)
  - Hence our plaintext must first be encoded as a number
    - Make sure the block size b is such that $10^{2b} < p < 10^{2b+2}$
    - If p is large enough why not use the printable ASCII-values minus 30...
    - ...and create a dinome sequence from each plaintext block
    - Maybe even add an EOL-token with code 01
    - Interpret this dinome sequence as a decimal number $a$

$$\boxed{\mathcal{E}(a) \equiv a^e \pmod{p}}$$

# Pohlig-Hellman baby example for small p

Let us consider a small example[1] with $e_{\mathcal{E}} = 7$ and $p = 11$,

meaning we can't encode more than 10 symbols:

| $a$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{E}(a)$ | 1 | 7 | 9 | 5 | 3 | 8 | 6 | 2 | 4 | 10 |

| $b$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}(b)$ | 1 | 8 | 5 | 9 | 4 | 7 | 2 | 6 | 3 | 10 |

This turns out to be equivalent to encryption with $e_{\mathcal{D}} = 3$ and $p = 11$:

[1] So in this case we can't use the blocksize (b=0) of the previous slide

# Relation between $e_{\mathcal{E}}$ and $e_{\mathcal{D}}$

- The needed property is $\mathcal{D}(\mathcal{E}(x)) = x$
- That translates to $(a^{e_{\mathcal{E}}})^{e_{\mathcal{D}}} \equiv a \pmod{p}$
- Or $a^{e_{\mathcal{E}} \cdot e_{\mathcal{D}}} \equiv a \pmod{p}$
- Fermat's little theorem comes to the rescue
  - We know that $a^{p-1} \equiv 1 \pmod{p}$
  - Therefore we want $e_{\mathcal{E}} \cdot e_{\mathcal{D}} \equiv 1 \pmod{p-1}$
- This works for our example since $7 \cdot 3 \equiv 1 \pmod{10}$

# Relaxation for $p$

- We can relax the conditions on $p$
- There is no need for $p = N$ to be prime
  - But we need the more general Euler-Fermat property
  - We know that $a^{\varphi(N)} \equiv 1 \pmod{N}$,

    for all $a$ with $\gcd(a, N) = 1$
  - There are fewer[2] possible plaintext options for $a$
- We need again that $e$ is relatively prime to $\varphi(N)$
  - So that we can find a $d$ with $e \cdot d \equiv 1 \pmod{\varphi(N)}$

[2] Although decryption still works (magically?) if $N$ is a product of different primes

# Known plaintext and the discrete logarithm problem

- In the Pohlig-Hellman symmetric scheme $e$ is a shared secret
- What problem do we need to solve to mount

  a known plaintext attack?
- Suppose we know $a$ (plaintext) and $b = a^e \pmod{N}$
- Finding $e$ is called a **discrete log(arithm) problem**
- You want to find $e = \log_a b \pmod{N}$
- In general this turns out to be a (very) hard problem

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - **Exponentiation ciphers with public exponent**
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# RSA (Rivest-Shamir-Adleman)

- This is like Pohlig-Hellman, but with a public exponent $e$
- In order to decrypt we need a $d$ which inverts $e$ modulo $\varphi(N)$
- This would be easy to calculate if we know $\varphi(N)$
- But $\varphi(N)$ depends on a factorization of $N$
- Just like discrete logs, factorization seems to be a **hard problem**

# RSA (danger: textbook variant)

## Its definition

### Definition (RSA)

RSA works with public information (Uppercase)

and private information (lowercase)

- A public modulus $N = pq$, which is the product of two private (secret) primes.

  Notice that $\varphi(N) = (p-1)(q-1)$, which is (as far as we know) hard to calculate

  if you do not know the primes p and q.
- A public exponent $E \in \mathbb{Z}^*_{\varphi(N)}$.
- A private exponent $d$ such that $Ed \equiv 1 \pmod{\varphi(N)}$.

  Note that $d$ can easily be calculated using Euclid's algorithm.
- $(N, E)$ is called the public key.
- $(p, q = \frac{N}{p}, d)$ is called the private key.
- $(N = pq, E, d)$ is called a public/private key "pair".

# RSA (Textbook variant)

## Its principle of operation

**Theorem**

*A message is represented as a positive $m < N$. This message is encoded as $C = m^E \pmod{N}$. Then $m \equiv C^d \pmod{N}$.*

**Proof.**

Let $C = m^E \pmod{N}$ and $Ed = 1 + k\varphi(N)$.

Then

$$\begin{aligned} C^d &\equiv (m^E)^d \pmod{N} \equiv m^{Ed} \pmod{N} \\ &\equiv m^{(1+k\varphi(N))} \pmod{N} \\ &\equiv m(m^{\varphi(N)})^k \pmod{N} \\ &\equiv m1^k \pmod{N} \equiv m \pmod{N} \end{aligned}$$

□

Who spots the (minor) omission in this proof?

# RSA (Textbook variant)

## Its principle of operation

**Theorem**

*A message is represented as a positive* $m < N$. *This message is encoded as* $C = m^E \pmod N$. *Then* $m \equiv C^d \pmod N$.

**Proof.**

Let $C = m^E \pmod N$ and $Ed = 1 + k\varphi(N)$.

Then

$$\begin{aligned} C^d &\equiv (m^E)^d \pmod N \equiv m^{Ed} \pmod N \\ &\equiv m^{(1+k\varphi(N))} \pmod N \\ &\equiv m(m^{\varphi(N)})^k \pmod N \\ &\equiv m1^k \pmod N \equiv m \pmod N \end{aligned}$$

□

Who spots the (minor) omission in this proof?

One should also consider the case where $\gcd(m, N) > 1$

# Outline

1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# DH (Diffie-Hellman)

- Consider an exponentiation cipher with a **known**, **fixed** message
- Both sides choose **their own** secret exponent...
- ... and communicate the cryptograms
- What good can this possibly be?
- The cryptograms enable both parties to compute a **shared secret**
  - which is again an encryption of the fixed message

    by an **unknown secret**
- The shared secret can now be used by both parties to be the

  secret key in Pohlig-Hellman or any other encryption scheme

# Diffie-Hellman

## Its definition

Let P be a prime and G a primitive root (or generator) of the group

$$\mathbb{Z}_P^* = \{G^0 = 1 = G^{P-1}, G^1 = G, G^2, G^3, \ldots, G^{P-2}\}$$

**Definition (Diffie-Hellman)**

Let two parties, say A and B, choose positive secret numbers $x, y < P - 1$

A publishes $X = G^x \pmod P$ and B publishes $Y = G^y \pmod P$.

The two parties now have a shared secret: $G^{xy} \pmod P$.

A knows $G^{xy} \equiv (G^y)^x \equiv Y^x \pmod P$

B knows $G^{xy} \equiv (G^x)^y \equiv X^y \pmod P$

Nobody, except A and B, knows $G^{xy} \pmod P$

Nobody knows $xy$ or even $xy \pmod P$, which is the unknown secret

# Outline

1 Numbers and basic arithmetic laws
- The rationals
- Primes
- Arithmetic in finite structures
- Euler's $\varphi$

2 Applications to cryptography
- Exponentiation ciphers with secret exponent
- Exponentiation ciphers with public exponent
- Exponentiation ciphers for key exchange

3 Calculating with polynomials
- Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
- Finite fields or Galois fields
- Application to AES(Rijndael)

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Addition and multiplication of polynomials (1)

- In $\mathbb{F}[X]$ one can add and multiply polynomials as usual
- Over $\mathbb{Q}$ we have

$$(X^2 - 3X + 4) + (X^3 - X^2 + 2X - 6) \quad = \quad X^3 - X - 2$$

$$(X^2 - 3X + 4) \cdot (X^3 - X^2 + 2X - 6) =$$
$$X^5 - 4X^4 + 9X^3 - 16X^2 + 26X - 24$$

# Addition and multiplication of polynomials (2)

- Over $\mathbb{Z}_2$ we have

$$(X^2 + X + 1) + (X^2 + 1) = X$$

$$(X^2 + X + 1) \cdot (X^2 + 1) = X^4 + X^3 + X + 1$$

- Over $\mathbb{Z}_3$ we have

$$(X^2 + X + 1) + (X^2 + 1) = 2X^2 + X + 2 = -X^2 + X - 1$$

$$\begin{aligned}(X^2 + X + 1) \cdot (X^2 + 1) &= X^4 + X^3 + 2X^2 + X + 1 \\ &= X^4 + X^3 - X^2 + X + 1\end{aligned}$$

# Example of Euclidean division

**Examples (reduction modulo the polynomial $X^2 + X + 1$ over $\mathbb{Q}$)**

$$
\begin{aligned}
X^3 + 3X - 4 &= X(X^2 + X + 1) - X^2 + 2X - 4 \\
&= X(X^2 + X + 1) - 1(X^2 + X + 1) + 3X - 3 \\
&= (X - 1)(X^2 + X + 1) + 3X - 3
\end{aligned}
$$

So $X^3 + 3X - 4 \equiv 3X - 3 \pmod{X^2 + X + 1}$

$X - 1$ is the quotient and $3X - 3$ is the remainder

# The equivalents of the primes in $\mathbb{F}[X]$

## Definition

A polynomial $g$ is called **irreducible** in $\mathbb{F}[X]$ if there are no lower degree ($> 0$) polynomials $h$ and $k$ such that $g = hk$.

The irreducible polynomials are the equivalents of the primes

## Theorem

*If g is irreducible then*

$$\mathbb{F}[X]/(g) = \{f \pmod g \mid f \in \mathbb{F}[X]\}$$

*is a field.*

# Examples of (ir)reducible polynomials

### Examples (of reducible polynomials)

- $X^2 - 3X + 2$ is reducible in $\mathbb{Q}[X]$
- $X^2 + 1$ is reducible over $\mathbb{Z}_2$
- $X^2 + X + 1$ is reducible over $\mathbb{Z}_3$

### Examples (of irreducible polynomials)

- $X^2 + 1$ is irreducible in $\mathbb{Q}[X]$
- $X^2 + 1$ is irreducible over $\mathbb{Z}_3$
- $X^2 + X + 1$ is irreducible over $\mathbb{Z}_2$

### Examples (Algebraic Number Fields, using $\mathbb{Q}$)

- $X^2 - 2$ is irreducible in $\mathbb{Q}[X]$ and
  $\mathbb{Q}[X]/(X^2 - 2) \cong \mathbb{Q}(\sqrt{2})$
- $X^2 + X + 1$ is irreducible in $\mathbb{Q}[X]$ and
  $\mathbb{Q}[X]/(X^2 + X + 1) \cong \mathbb{Q}(-1/2 + 1/2i\sqrt{3})$

# Outline


1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Irreducibles over $\mathbb{Z}_p$

## Theorem

- *Taking $\mathbb{F} = \mathbb{Z}_p$ for a prime p and g an irreducible polynomial of degree n with coefficients in $\mathbb{F}$, $\mathbb{F}[X]/(g)$ is a field with $p^n$ elements.*
- *For any prime p and natural number $n > 0$ there is exactly one field, denoted $GF(p^n)$ or $\mathbb{F}_{p^n}$, with $p^n$ elements (up to isomorphism).*

In honour of Évariste Galois these finite fields are also called Galois fields.

Uniqueness up to isomorphism tells you it doesn't matter which irreducible polynomial is used for the construction.

# Properties of finite fields

## and their cyclic multiplicative subgroups

### Theorem

*For any finite field* $\mathbb{F}$

- $|\mathbb{F}| = p^n$, *where p is a prime called the characteristic of* $\mathbb{F}$, *being the smallest number for which the p-time repetition* $1 + 1 + \ldots + 1$ *is equal to 0.*
- *The multiplicative group of* $\mathbb{F}$, *which is* $\mathbb{F}\backslash\{0\}$, *is always cyclic*
- *The irreducible polynomial g is called* *primitive* *if* $\alpha = X \pmod{g}$ *is a generator of this (cyclic) multiplicative group*
- $GF(p) \cong \mathbb{Z}_p$
- *For* $n > 1$*:* $GF(p^n) \ncong (\mathbb{Z}_p)^n$
- *For* $n > 1$*:* $GF(p^n) \ncong \mathbb{Z}_{p^n}$

# Examples of finite fields and primitive polynomials

## Examples

- $X^2 + X + 1$ is (irreducible and) primitive over $GF(2)$
- $GF(4) = GF(2^2) = \mathbb{Z}_2[X]/(X^2 + X + 1) =$

  $\{0, 1, \alpha, \alpha^2 = \alpha + 1\}$ with generator $\alpha = X \pmod{X^2 + X + 1}$.
- $X^2 + 1$ is irreducible, but not primitive over $GF(3)$
- $X^2 + 2X + 2$ is (irreducible and) primitive over $GF(3)$
- $GF(9) = GF(3^2) = \mathbb{Z}_3[X]/(X^2 + 2X + 2) =$

  $\{0, \alpha, \alpha^2 = \alpha + 1, \alpha^3 = 2\alpha + 1, \alpha^4 = 2,$

  $\alpha^5 = 2\alpha, \alpha^6 = 2\alpha + 2, \alpha^7 = \alpha + 2, \alpha^8 = 1\},$

  with generator $\alpha = X \pmod{X^2 + 2X + 2}$.

# Outline

1. Numbers and basic arithmetic laws
   - The rationals
   - Primes
   - Arithmetic in finite structures
   - Euler's $\varphi$
2. Applications to cryptography
   - Exponentiation ciphers with secret exponent
   - Exponentiation ciphers with public exponent
   - Exponentiation ciphers for key exchange
3. Calculating with polynomials
   - Polynomials over $\mathbb{F}$ ($\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime $p$)
   - Finite fields or Galois fields
   - Application to AES(Rijndael)

# Use of Galois Fields in AES (1)

- The *S-box* uses polynomials over $GF(2)$
  - The inverse modulo the irreducible $x^8 + x^4 + x^3 + x + 1$
  - Multiplication by $x^4 + x^3 + x^2 + x + 1$ modulo the (reducible) $x^8 + 1$
  - Addition of $x^6 + x^5 + x + 1$ also modulo the (reducible) $x^8 + 1$
- *MixColumn* uses polynomials with coefficients over $GF(2^8)$ modulo the reducible polynomial $x^4 + \mathbf{01}$
  - $GF(2^8) \cong \mathbb{Z}_2[x]/(x^8 + x^4 + x^3 + x + 1)$, represented by hex digits **XY**
  - Multiplication by $\mathbf{03}x^3 + \mathbf{01}x^2 + \mathbf{01}x + \mathbf{02}$

    and for the inverse by $\mathbf{0B}x^3 + \mathbf{0D}x^2 + \mathbf{09}x + \mathbf{0E}$

# Use of Galois Fields in AES (2)

- *Key expansion* uses
  - Arithmetic in $GF(2^8)$ for generating the constants $C_i = x^{i-1}$ working modulo the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$
  - The polynomial $x^3$ modulo $x^4 + \mathbf{01}$ over $GF(2^8)$ for rotations of columns

For a concise treatment of Rijndael (AES) for algebraists by Hendrik Lenstra, see

`https://www.math.berkeley.edu/~hwl/papers/rijndael0.pdf`

# Classical Cryptography

## Block ciphers: DES and AES

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.3, 2020/03/04 14:52:11 UTC)

Thursday, March 5, 2020

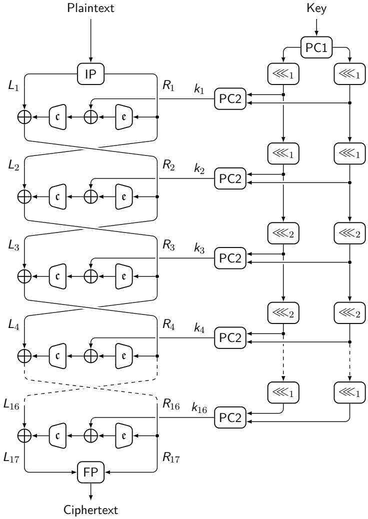
1. The Data Encryption Standard DES

2. The Advanced Encryption Standard AES

# Outline


1. The Data Encryption Standard DES
2. The Advanced Encryption Standard AES

# DES Overview



# DES initial (IP) and final (FP) permutation

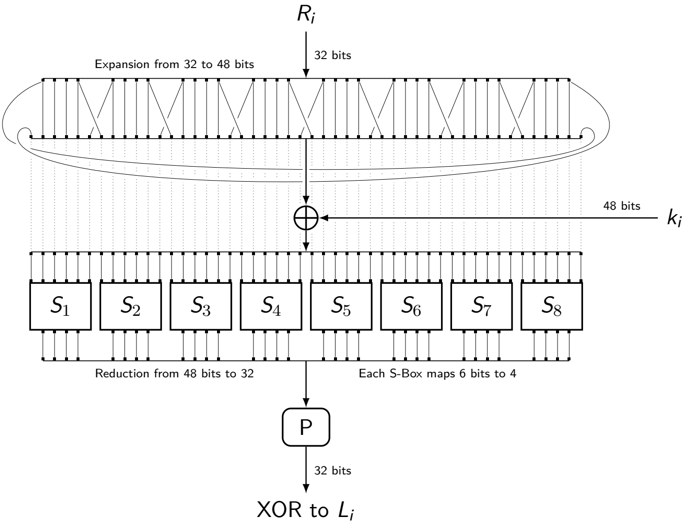
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Read this table row by row

For IP bit 58 moves to the first position

For FP bit 1 moves to position 58

# DES round



$R_i$
32 bits
Expansion from 32 to 48 bits
48 bits
$k_i$
$S_1$
$S_2$
$S_3$
$S_4$
$S_5$
$S_6$
$S_7$
$S_8$
Reduction from 48 bits to 32
Each S-Box maps 6 bits to 4
P
32 bits
XOR to $L_i$

# DES P-box permutation

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

# DES PC1 (permuted choice 1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Reduces from 64 bits to 56 bits by leaving out parity bits

# DES PC2 (permuted choice 2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 |
| 15 | 6 | 21 | 10 | 23 | 19 | 12 | 4 |
| 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 |
| 51 | 45 | 33 | 48 | 44 | 49 | 39 | 56 |
| 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

Reduces from 56 bits to 48 bits leaving out bits

9, 18, 22, 25, 35, 38, 43, 54

# Outline

1. The Data Encryption Standard DES
2. The Advanced Encryption Standard AES

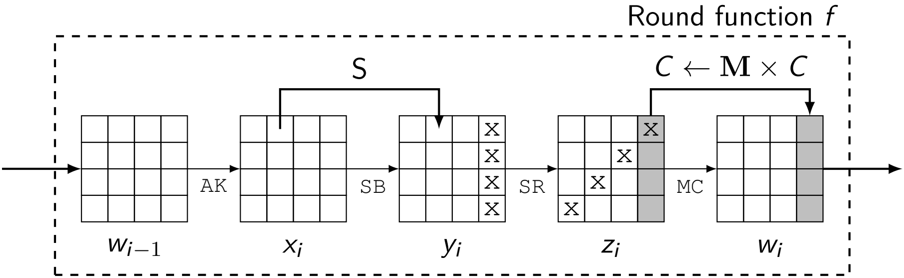
# AES state

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

Each square represents one byte for a total of 128 bits

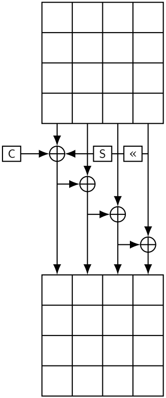
Each column represents a word consisting of 4 bytes

# AES Overview



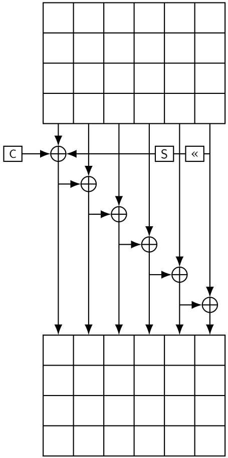

AK: Add Key, SB: Sub Bytes, SR: Shift Rows, MC: Mix Columns

# AES key schedule (128 bit keys)

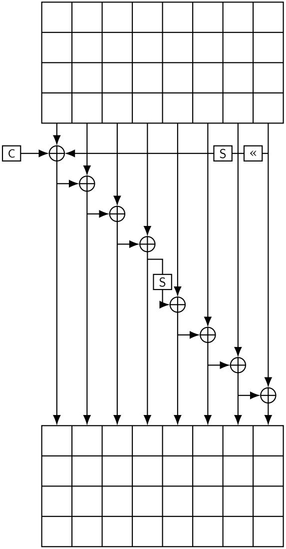



C: (Round dependent) Constant, S: Sub word, «: Rotate word

# AES key schedule (192 bit keys)



# AES key schedule (256 bit keys)



# AES animation and stick guide

`https://www.youtube.com/watch?v=gP4PqVGudtg`

`http://www.moserware.com/2009/09/`
`stick-figure-guide-to-advanced.html`

# Classical Cryptography

## (Linear )Feedback Shift Registers

Karst Koymans

Informatics Institute

University of Amsterdam

(version 19.3, 2020/03/09 12:59:04)

Monday, March 9, 2020

# Table of Contents

1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# Outline


1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# Unusual (but better) convention

**Warning**

These slides use a representation which shifts feedback registers to the left and not to the right

# Outline


1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# Using sequences instead of registers

- We can consider, just as with the Feistel scheme, an infinite sequence of values and no register
- Mathematically this is nicer
  - Useful for the best specification
- Computationally a shift register is nicer
  - Useful for an efficient implementation

# Register based feedback function

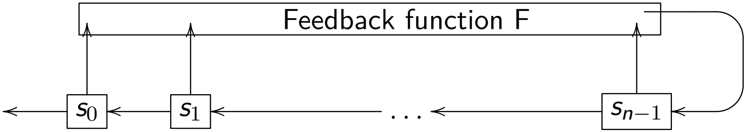



Figure 1: Register based feedback function

Next state calculation (implementation)

$$s_i(t+1) = s_{i+1}(t) \qquad \text{for all } i < n-1 \text{ and time } t \geq 0$$

$$s_{n-1}(t+1) = F(s_0(t), \ldots, s_{n-1}(t)) \qquad \text{for all } t \geq 0$$

# "Feedback" function with an infinite sequence

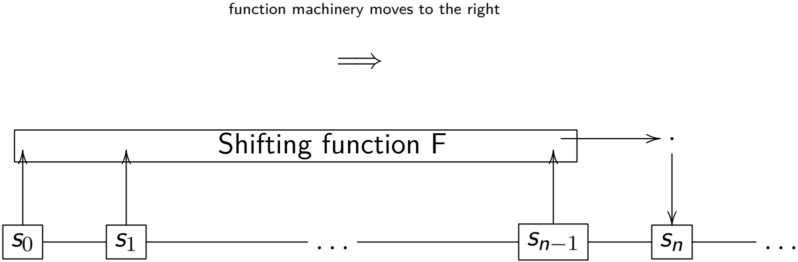



Figure 2: Sequence based shifting function

A recursive sequence specification ("Fibonacci style")

$$s_{n+k} = F(s_k, \ldots, s_{n+k-1}) \text{ for all } k \geq 0$$

$$(\text{or } s_k = F(s_{k-n}, \ldots, s_{k-1}) \text{ for all } k \geq n)$$

# Example feedback function

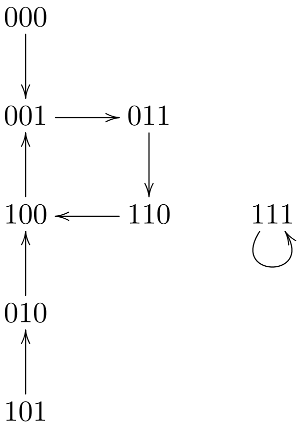



Figure 3: Cycle structure for feedback function $F(s_0, s_1, s_2) = s_0 \cdot s_2 \oplus s_1 \oplus 1$

# Maximum length cycle feedback function

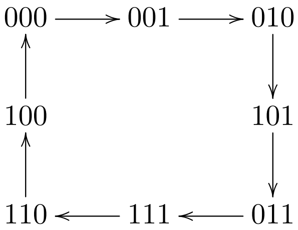



Figure 4: Cycle structure for feedback function $F(s_0, s_1, s_2) = s_0 \oplus s_2 \oplus s_1 \cdot s_2 \oplus 1$

# From graph to boolean formula

| $s_0 s_1 s_2$ | $F(s_0, s_1, s_2)$ | Component |
|---|---|---|
| 000 | 1 | $(s_0 \oplus 1) \cdot (s_1 \oplus 1) \cdot (s_2 \oplus 1)$ |
| 001 | 0 | - |
| 010 | 1 | $(s_0 \oplus 1) \cdot s_1 \cdot (s_2 \oplus 1)$ |
| 011 | 1 | $(s_0 \oplus 1) \cdot s_1 \cdot s_2$ |
| 100 | 0 | - |
| 101 | 1 | $s_0 \cdot (s_1 \oplus 1) \cdot s_2$ |
| 110 | 0 | - |
| 111 | 0 | - |

Figure 5: Components to be added for given cycle structure

# Outline


1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# Linear feedback function with register

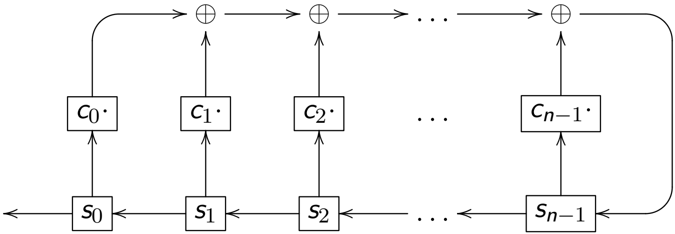



Figure 6: Linear feedback in Fibonacci mode

Next state calculation (implementation)

$$s_i(t+1) = s_{i+1}(t) \quad \text{for all } i < n-1 \text{ and time } t \geq 0$$

$$s_{n-1}(t+1) = c_0 \cdot s_0(t) \oplus \ldots \oplus c_{n-1} \cdot s_{n-1}(t) \quad \text{for all } t \geq 0$$

# Linear shifting function for sequence

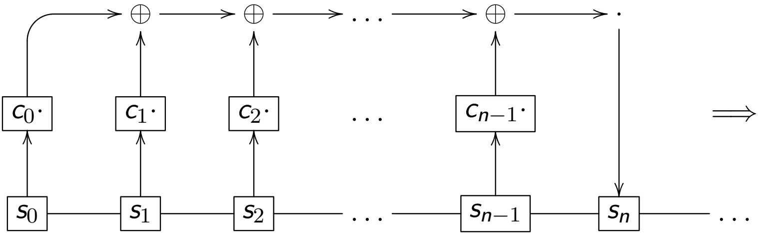



Figure 7: Linear recursion in Fibonacci mode

A recursive sequence specification ("Fibonacci style")

$$s_{n+k} = c_0 \cdot s_k \oplus \ldots \oplus c_{n-1} \cdot s_{n+k-1} \text{ for all } k \geq 0$$

$$(\text{or } s_k = c_0 \cdot s_{k-n} \oplus \ldots \oplus c_{n-1} \cdot s_{k-1} \text{ for all } k \geq n)$$

# Symmetric representation feedback mechanism

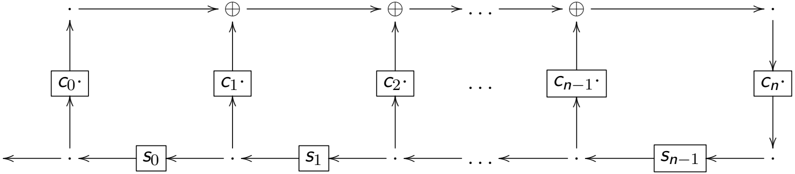



Figure 8: More symmetric figure

- Note that we assume we always have $c_0 = 1$ and $c_n = 1$; why?

# Symmetric representation feedback mechanism

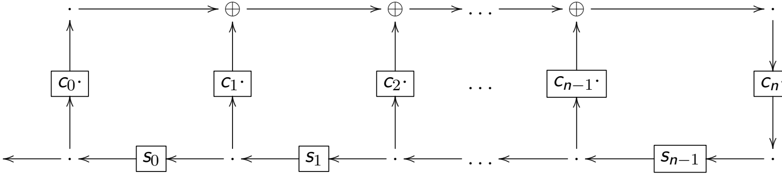



Figure 8: More symmetric figure

- Note that we assume we always have $c_0 = 1$ and $c_n = 1$; why?
  - If $c_0 = 0$ we would have a shorter LFSR

# Symmetric representation feedback mechanism

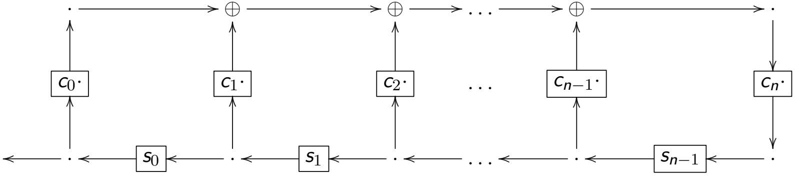



Figure 8: More symmetric figure

- Note that we assume we always have $c_0 = 1$ and $c_n = 1$; why?
  - If $c_0 = 0$ we would have a shorter LFSR
  - If $c_n = 0$ the feedback would always be 0

# Symmetric representation feedback mechanism

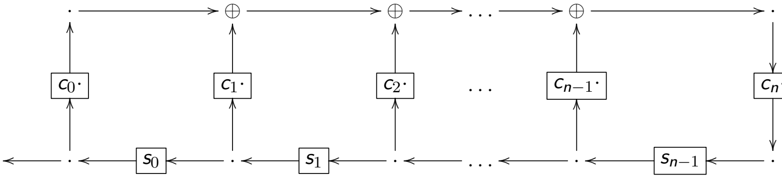



Figure 8: More symmetric figure

- Note that we assume we always have $c_0 = 1$ and $c_n = 1$; why?
  - If $c_0 = 0$ we would have a shorter LFSR
  - If $c_n = 0$ the feedback would always be 0
- The $c_i\cdot$ multiplies by 0 or 1, so in the sequence of xor operations
  - $s_i$ is absent if $c_i = 0$
  - $s_i$ is present if $c_i = 1$

# Characteristic and feedback polynomials

- We call $\sum_{i=0}^{n} c_i X^i$ the **characteristic** polynomial ($\chi$)
- We call $\sum_{i=0}^{n} c_i X^{n-i}$ the **feedback** polynomial ($\phi$)
- $\phi(X) = X^n \chi(1/X)$
- $\chi(X) = X^n \phi(1/X)$

**Theorem**

*The LFSR has a maximum period of length $2^n - 1$ iff the characteristic (equivalently the feedback) polynomial is* ***primitive****.*

# Reducible feedback polynomial $X^3 + X^2 + X + 1$

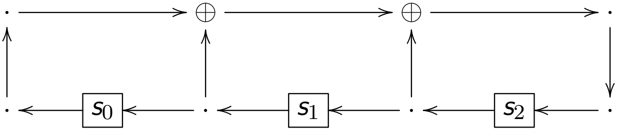
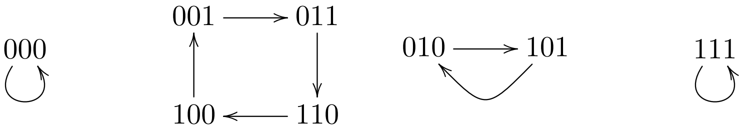



Figure 9: This LFSR does not generate a maximal sequence



# Primitive feedback polynomial $X^3 + X^2 + 1$

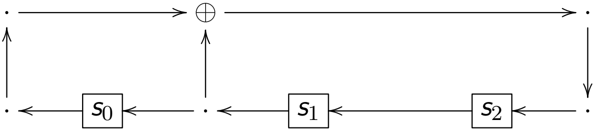
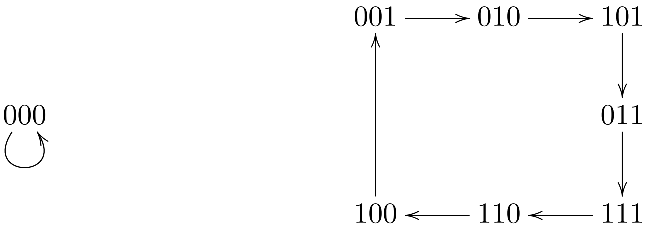



Figure 10: This LFSR does generate a maximal sequence



# LFSR properties

- Can be used as a **DRBG** (Deterministic Random Bit Generator)
  - This is also called a **PRNG** (PseudoRandom Number Generator)
- Can not be used by itself for cryptographic purposes
  - A known plaintext attack reveals the internal state
  - From there the whole future and history of the state can be determined
- Can be combined with non-linear methods to produce a **CSPRNG**
  - Trivium is a stream cipher that is based on
    such a Cryptographically Secure PRNG
- An LFSR with the maximum period satisfies the **Golomb criteria**

# Golomb criterion 1

- The numbers of zeroes and ones are **balanced**
  - Since the length of the periodic sequence $(2^n - 1)$ is odd this can't be true exactly
  - There will be $2^{n-1} - 1$ zeroes and $2^{n-1}$ ones
    - "One more one"

# Golomb criterion 2

- Let a **run** be a sequence of consecutive zeroes (or ones)

  that is not a part of a longer sequence of zeroes (or ones)
- Longer runs should occur less often
  - There are $2^{n-k-2}$ runs of zeroes (and also of ones) of length $k < n-1$
    - So in total there are $2^{n-k-1}$ runs of length $k < n-1$
  - There are two special cases ($k = n-1$ and $k = n$)
    - A run of $n-1$ zeroes (there is no run of $n$ zeroes)
    - A run of $n$ ones (there is no run of $n-1$ ones)

# Golomb criterion 3

- **Autocorrelation** should be constant (near 0)
- $Corr(\delta) = 1 - \frac{2}{p} \sum_{i=0}^{p-1} (s_i \oplus s_{i+\delta})$
  - where $p = 2^n - 1$
  - Note that $Corr(0) = 1$
- Now use that xoring two sequences that satisfy a linear recurrence gives another sequence that does so
- It follows that $\sum_{i=0}^{p-1} (s_i \oplus s_{i+\delta}) = 2^{n-1}$ being the number of ones in such a sequence
- We conclude $Corr(\delta) = 1 - \frac{2}{2^n - 1} 2^{n-1} = -\frac{1}{2^n - 1} = -\frac{1}{p}$

# Outline


1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# Inverted (Galois) feedback mechanism (1)

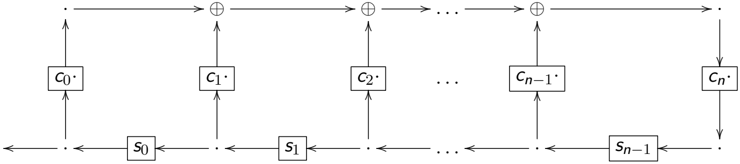



Figure 11: Linear feedback in Fibonacci mode

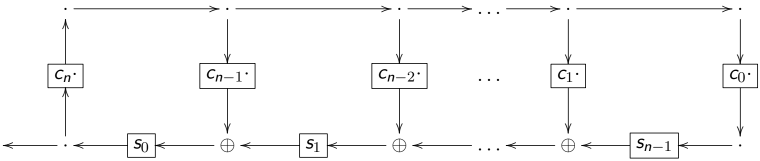



Figure 12: Linear feedback in Galois mode

# Inverted (Galois) feedback mechanism (2)

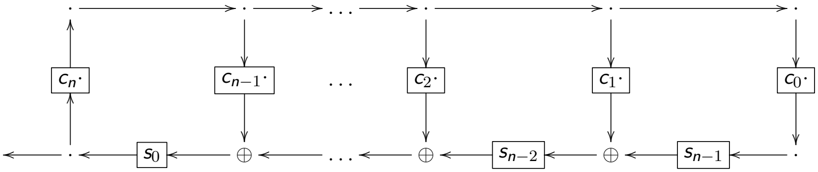



Figure 13: Linear feedback in Galois mode

- Note the **inverted order** of the constants $c_i$
- In other words you could say that the roles of characteristic and feedback polynomial are reversed
- This implementation is more time efficient since the xors can be parallellized

# Galois characteristic polynomial $X^3 + X + 1$

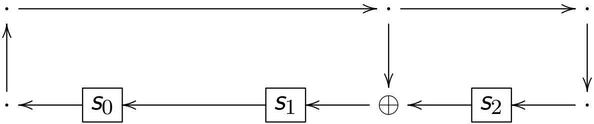
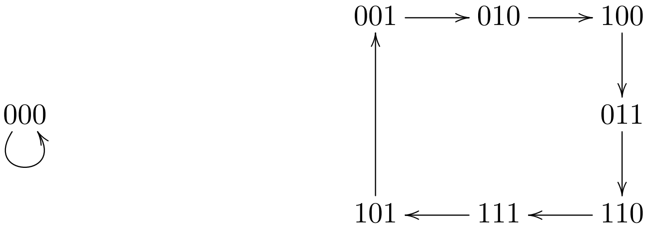



Figure 14: This LFSR does generate a maximal sequence



# Comparison of Fibonacci and Galois (same stream)

- Fibonacci with feedback polynomial $\phi(X) = X^3 + X^2 + 1$
- Galois with characteristic polynomial $\chi(X) = X^3 + X + 1$

| Fibonacci | | | $\mathbb{F}_{2^3} = \mathbb{F}_2[X]/(\chi) = \mathbb{F}_2[\alpha]$ | Galois | | |
|---|---|---|---|---|---|---|
| **0** | 0 | 1 | $1$ | **0** | 0 | 1 |
| **0** | 1 | 0 | $\alpha$ | **0** | 1 | 0 |
| **1** | 0 | 1 | $\alpha^2$ | **1** | 0 | 0 |
| **0** | 1 | 1 | $\alpha^3 = \alpha + 1$ | **0** | 1 | 1 |
| **1** | 1 | 1 | $\alpha^2 + \alpha$ | **1** | 1 | 0 |
| **1** | 1 | 0 | $\alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$ | **1** | 1 | 1 |
| **1** | 0 | 0 | $\alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1$ | **1** | 0 | 1 |

# Outline


1. Introduction to feedback shift registers
2. General framework for feedback shift registers
3. Linear feedback shift registers (LFSRs), Fibonacci style
4. Linear feedback shift registers (LFSRs), Galois style
5. The Trivium stream cipher

# A combination of linear and nonlinear feedback

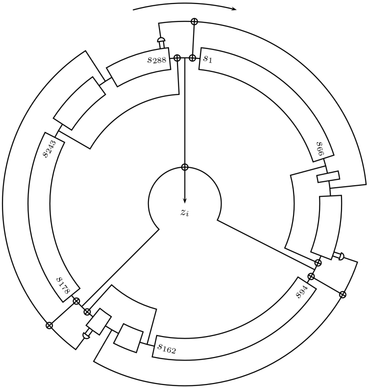



Figure 15: A circular set of three shift registers

# The Trivium cipher feedback loop

$$
\begin{aligned}
t_1 &:= s_{66} + s_{93} \\
t_2 &:= s_{162} + s_{177} \\
t_3 &:= s_{243} + s_{288} \\
z_i &:= t_1 + t_2 + t_3 \\
t_1 &:= t_1 + s_{91} \cdot s_{92} + s_{171} \\
t_2 &:= t_2 + s_{175} \cdot s_{176} + s_{264} \\
t_3 &:= t_3 + s_{286} \cdot s_{287} + s_{69} \\
(s_1, s_2, \ldots, s_{93}) &:= (t_3, s_1, \ldots, s_{92}) \\
(s_{94}, s_{95}, \ldots, s_{177}) &:= (t_1, s_{94}, \ldots, s_{176}) \\
(s_{178}, s_{179}, \ldots, s_{288}) &:= (t_2, s_{178}, \ldots, s_{287})
\end{aligned}
$$

Figure 16: The Trivium cipher recursions (with output $z_0, z_1, \ldots$)

# The Trivium cipher key and IV setup

$$
\begin{aligned}
(s_1, s_2, \ldots, s_{93}) &:= (K_1, \ldots, K_{80}, 0, \ldots, 0) \\
(s_{94}, s_{95}, \ldots, s_{177}) &:= (IV_1, \ldots, IV_{80}, 0, \ldots, 0) \\
(s_{178}, s_{179}, \ldots, s_{288}) &:= (0, \ldots, 0, 1, 1, 1)
\end{aligned}
$$

Figure 17: The Trivium cipher initialization

This is followed by 4 full cycles of the feedback loop

before the stream cipher generates output.