

Mathematical Puzzles

Prof. Dr. Th. Risse

An amusing,
brisk and cool,
enriching and entertaining,
informative and oriented towards practical applications,
playful,
relevant and rewarding,
stimulating,
thought-provoking
little contribution to the (general) mathematical education!

Table of Contents

0. Introduction

1. Riddles, s. e.g. [7]

- Measuring with Two Jugs
- Races
- Census and its Boycott
- Zig-Zag between Trains
- Outward and Return Journey
- Magic Squares
- Conspicuous Text
- No Talk about Money
- Corrupt Postal System
- Equal Opportunities
- Two and More Eyes

2. Prime Numbers

- Fermat-Numbers
- Euler-Numbers
- Mersenne-Numbers

3. Computations with Remainders

- Crucial is What is Left Over
- Computing With Remainders
- Adroit Computing With Remainders
- Euclid & little Fermat
- Fermat, Euler and More
- Chinese Stuff
- Galois Fields $\mathbb{GF}(p)$
- Galois Fields $\mathbb{GF}(p^n)$

4. Cryptography

- Caesar and Cohorts • Caesar in General • Vigenère and Accomplices • Permutations • DES • Public Keys? • RSA
- AES • Elliptic Curves over \mathbb{R} • Elliptic Curves over $\mathbb{GF}(p)$
- Elliptic Curves over $\mathbb{GF}(2^m)$ • Elliptic Curve Cryptography, ECC

5. Compression

- Exploiting Relative Frequencies • Using Dictionaries

6. Probability & Intuition

- Cards & Goats • Algorithms to Generate Chance? • What is Randomness?

7. Sources and Links

Solutions to Problems

0. Introduction

To begin with You'll find some mathematical riddles. But there is more serious stuff. Several *algorithms* to be tried are provided by this document to explore procedures of cryptography, coding, probability, etc.

There are other in this sense *interactive documents*, e.g.

www.weblearn.hs-bremen.de/risse/MAI/docs/numerics.pdf or
www.weblearn.hs-bremen.de/risse/MAI/docs/heath.pdf (German)

The functionality of pdf-documents provides

convenient selection of **problem areas** of interest or of single problems • and, uniquely, **execution of algorithms**

easy navigation between **problem** and **solution** and vice versa,

simple visit of the numerous **links** to informations on our webDAV server or in the WWW.

1. Riddles, s. e.g. [7]

• Measuring with Two Jugs

PROBLEM 1. There are two jugs at hand with a capacity of $p\ell$ and $q\ell$ liters and any amount of water.

What quantities m of water can be measured out?

(a) $p = 5, q = 3, m = 4$

(b) $p = 5, q = 3, m = 1$

(c) $p = 4, q = 9, m = 1, 2, \dots, 13$

(d) $p = 6, q = 3, m = 4$

- **Races**

PROBLEM 2.

- (a) Climbing a 3000m mountain top Sisyphos makes 300m a day only to loose 200m each night again.
Wenn does Sisyphos reach the top?
- (b) At a 100m race the first runner A beats the second B by 10m, and the second B beats the third C by 10m.
How many meters is the first runner A ahead of the third C when crossing the finishing line?

- **Census and its Boycott**

PROBLEM 3.

(a) At a census there is the following dialog:

Field helper: number of children?

Citizen: three!

Field helper: age of Your children in whole numbers?

Citizen: The product of the years is 36.

Field helper: This not a sufficient answer!

Citizen: The sum of the ages equals the
number of the house of our next neighbour.

(Field helper acquires the number.)

Field helper: That is still not a sufficient answer!

Citizen: Our eldest child plays the piano.

How old are the three children?

- **Zig-Zag between Trains**

PROBLEM 4.

- (a) Two trains start on the same line 100km apart to drive at 50km/h towards each other. A fly flies from one to the other at 75km/h. How many kilometres has the fly travelled up to its unavoidable fate?

- **Outward and Return Journey**

PROBLEM 5.

(a) In A somebody gets up at sunrise and walks with many rests to B where he arrives at sunset.

The next day he walks back on the same route, again pausing a bit here and there.

There is a point of the route the roamer at the same time of day hits both on the outward as on the return journey.

- **Magic Squares**

PROBLEM 6.

- (a) Magic squares are natural numbers arranged in a square grid, i.e. a quadratic matrix, such that the sum of all numbers in each row, in each column, and in each diagonal are all the same!

a	b	c
d	e	f
g	h	i

mit
$$\begin{aligned} a + b + c &= s \dots \\ a + d + g &= s \dots \\ a + e + i &= s \dots \end{aligned}$$

Taking symmetry into consideration, there is exactly one magic square consisting of the natural numbers 1, 2, ..., 9 arranged in a 3×3 -matrix.

- **Conspicuous Text**

PROBLEM 7.

- (a) Study this paragraph and all things in it. What is virtually wrong with it? Actually, nothing in it is wrong, but you must admit that it is most unusual. Don't zip through it quickly, but study it scrupulously. With luck you should spot what is so particular about it. Can you say what it is? Tax your brains and try again. Don't miss a word or a symbol. It isn't all that difficult.

- **No Talk about Money**

PROBLEM 8.

- (a) The boss in an office wants to acquire the average salary of his employees without getting to know individual salaries und thus breaking privacy. How does he proceed?

- **Corrupt Postal System**

PROBLEM 9.

- (a) In a corrupt postal system each letter is opened and the content stolen independently of its value. Only securely closed strong boxes are delivered reliably (because it takes too much hassle to open them).

How can Bob send a valuable item to Alice in some strong box which can be locked with several locks when they both can communicate about the transfer?

- **Equal Opportunities**

PROBLEM 10.

- (a) Alice and Bob live in different cities and decide to go to see each other in turns. They want to find out who starts to drive to the other by tossing a coin.

How do they find out if they live in different cities?

- **Two and More Eyes**

PROBLEM 11. It is called the *Two Eyes Principle* if two persons each with a separate key are necessary to open a treasure box, or if two passwords are necessary to open a file.

Each person opens her/his lock of the treasure box by her/his own key or adds her/his part of the password to complete the password.

(a) Alice, Bob and Claire own a treasure box with several locks. They want to make sure that only at least two persons together can get at the content of the treasure box.

How many locks and how many keys to each lock do they need?

(b) Now, Alice, Bob, Claire and Denis want to be sure that only at least two persons together can open the treasure box.

Minimally how many locks and minimally how many keys to each lock do they need?

(c) Only at least m persons together out of a total of n persons are meant to be able to open the treasure box.

How many locks, how many keys do they need?

2. Prime Numbers

In all modern cryptographical algorithms prime numbers play a decisive role. On top of that prime numbers challenged not only mathematicians for millennia and, (futile) attempts to generate prime numbers algorithmically date back centuries.

- **Fermat-Numbers**

PROBLEM 12. Fermat¹ numbers are specified by

$$F(n) = 2^{2^n} + 1$$

- (a) Fermat himself misleadingly believed to enumerate (all?) prime numbers in this way.

¹ Pierre Fermat (1601-1665)

- **Euler-Numbers**

PROBLEM 13. Euler² numbers are defined by

$$E(n) = n^2 - n + 41$$

(a) Only the first 40 Euler-numbers are prime.

² Leonhard Euler (1707-1783)

- **Mersenne-Numbers**

PROBLEM 14. Mersenne³ numbers are defined by

$$M(n) = 2^n - 1$$

(a) Only some Mersenne numbers are prime. But,

$$n \text{ not prime} \Rightarrow M(n) \text{ not prime}$$

Unfortunately, $M(n)$ is not necessarily prime if n is prime – as already a small ($< 2^{12}$) Mersenne number with four digits shows.

³Marin Mersenne (1588-1648)

3. Computations with Remainders

- **Crucial is What is Left Over**

Modulo-Arithmetic, i.e. computations with remainders, is essential (not only) in cryptography.

$$n \bmod m = r \iff n = vm + r \text{ für } n, v \in \mathbb{Z}, m, r \in \mathbb{N} \text{ und } 0 \leq r < m$$

PROBLEM 15.

- (a) Which day of the week do we have in n days?
- (b) Which day of the week did we have n days ago?
- (c) How is the UNIX-date computed, if an internal counter counts the seconds since **1.1.1970 0h** ?

• Computing With Remainders

$$n \equiv r \pmod{m} \iff m \mid (n - r) \iff m \mid n - r$$

$$n \equiv r \pmod{m} \iff n - r = v \cdot m \text{ für } m, r, v \in \mathbb{N} \text{ und } 0 \leq r < m$$

PROBLEM 16.

(a) Connection of $n \bmod m = r$ and $n \equiv r \pmod{m}$?

(b) **additivity, multiplicativity:**

$$\left. \begin{array}{l} n_1 \equiv r_1 \pmod{m} \\ n_2 \equiv r_2 \pmod{m} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} (n_1 \pm n_2) \equiv (r_1 \pm r_2) \pmod{m} \\ (n_1 \cdot n_2) \equiv (r_1 \cdot r_2) \pmod{m} \end{array} \right.$$

(c) **scalar multiples, powers**

$$n \equiv r \pmod{m} \Rightarrow \left\{ \begin{array}{l} c \cdot n \equiv c \cdot r \pmod{m} \text{ für jedes } c \in \mathbb{N} \\ n^p \equiv r^p \pmod{m} \text{ für jedes } p \in \mathbb{N} \end{array} \right.$$

(d) **transitivity**

$$r \equiv s \pmod{m}, s \equiv t \pmod{m} \Rightarrow r \equiv t \pmod{m}$$

• Adroit Computing With Remainders

Let $s(n) = \sum_{i=0}^{\infty} z_i$ denote the *cross sum* of $n = \sum_{i=0}^{\infty} z_i 10^i$.

PROBLEM 17. *Better to test dividability than to divide!*

- (a) $3 \mid s(n) \Rightarrow 3 \mid n$ as well as $9 \mid s(n) \Rightarrow 9 \mid n$
 Compute $1234567890 \bmod 3$, $1234567890 \bmod 9$ etc.
- (b) $11 \mid \sum_{i=0}^{\infty} (-1)^i z_i \Rightarrow 11 \mid \sum_{i=0}^{\infty} z_i 10^i$
 Compute $1234567890 \bmod 11$ etc.
- (c) The last digit of the 10-digit ISBNNumber is a check digit, an error checking number, namely $n \bmod 11$ if $n = \sum_{i=1}^9 i \cdot z_i$ denotes the weighed sum $1 \cdot z_1 + 2 \cdot z_2 + \dots + 9 \cdot z_9$ of the first nine digits $z_1 \dots z_9$.
(In case $n \bmod 11 = 10$ the check digit is represented by X.)
- (d) $7 \mid \sum_{i=0}^{\infty} (z_{7i+0} + 3z_{7i+1} + 2z_{7i+2} - z_{7i+3} - 3z_{7i+4} - 2z_{7i+5} + z_{7i+6})$
 $\Rightarrow 7 \mid \sum_{i=0}^{\infty} z_i 10^i$
 Compute $1234567890 \bmod 7$ etc.
- (e) Parity, ECC, CRC, RSC, ...?

• Euclid & little Fermat

PROBLEM 18. $\gcd(a, b)$ denotes *greatest common divisor*, \gcd of $a \in \mathbb{N}$ and $b \in \mathbb{N}$, i.e. $\gcd(a, b) = d \in \mathbb{N}$ with $d|a$ and $d|b$ as well as maximality, i.e. $d'|a, d'|b \Rightarrow d'|d$.

- (a) For $a, b \in \mathbb{N}$ holds $\gcd(a, b) = \gcd(a, b \bmod a) = \gcd(b, a \bmod b)$
- (b) By iteration we get the (terminating) Euclidean⁴ algorithm.
- (c) Fermat⁵'s Little Theorem, FLT: if p is prime then

$$a^{p-1} \equiv 1 \pmod{p}$$

for all $a \in \mathbb{N}$ with $\gcd(a, p) = 1$

Contraposition:

$$a^{n-1} \not\equiv 1 \pmod{n} \text{ for **one** } a \in \mathbb{N} \Rightarrow n \text{ is combined!}$$

- (d) The implication holds $n \text{ prim} \Rightarrow n | 2^{n-1} - 1$
but **not** its contraposition $n \text{ prim} \Leftarrow n | 2^{n-1} - 1$

⁴ Euclid of Alexandria (ca 325-265)

⁵ Pierre Fermat (1601-1665)

- **Fermat, Euler and More**

PROBLEM 19. The Euler⁶ function φ is defined by

$$\varphi(n) = |\{m \in \mathbb{N} : m < n, \gcd(m, n) = 1\}|$$

- (a) If p is prime then $\varphi(p) = p - 1$.
- (b) If p is prime then $\varphi(p^k) = p^k - p^{k-1} = (p - 1)p^{k-1}$.
- (c) If r and s relatively prime then $\varphi(r \cdot s) = \varphi(r) \cdot \varphi(s)$.
- (d) The prime factor decomposition of n provides a simple computation of $\varphi(n)$. Especially, for prime p and q we have

$$\varphi(p \cdot q) = \varphi(n) = n - (p + q) + 1 = (p - 1)(q - 1) \text{ für } n = p \cdot q$$

- (e) Theorem of Euler, EFT⁷:

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

for each $n \in \mathbb{N}$ and each a relatively prime to n .

⁶ Leonhard Euler (1707-1783)

⁷Euler-Fermat-Theorem, 1736

- Chinese Stuff

PROBLEM 20.

- (a) **Chinese Remainder Theorem:** Let $m_1, m_2, \dots, m_n \in \mathbb{N}$ be pairwise relatively prime. To find all solutions $x \in \mathbb{N}$ with

$$x \equiv r_i \pmod{m_i} \quad \text{für } i = 1, \dots, n$$

determine $m = \prod_{i=1}^n m_i$ and $b_i = m/m_i$ as well as x_i with $x_i b_i \equiv 1 \pmod{m_i}$, hence x_i as the (modulo m_i)-inverse to b_i for $i = 1, \dots, n$. Then:

$$x \equiv \sum_{i=1}^n (x_i b_i r_i) \pmod{m}$$

- (b) If p and q relatively prime, then

$$x = y \pmod{p} \text{ und } x = y \pmod{q} \quad \Rightarrow \quad x = y \pmod{pq}$$

- (c) The age of say party guests can be computed by the remainders when dividing the unknown age by 3, 5 and 7.

• **Galois Fields** $\mathbb{GF}(p)$

PROBLEM 21. Usually arithmetic takes place in fields with infinitely many elements, like \mathbb{Q} , \mathbb{R} and \mathbb{C} . However, in e.g. cryptography only fields with finitely many elements are relevant and hence needed.

As a reminder, a field is a set F of elements with two operations, namely addition $+$ and multiplication \cdot , so that $(F, +)$ (with zero-element 0) and $(F^*, \cdot) = (F \setminus \{0\}, \cdot)$ (with one-element 1) are commutative groups and the usual laws of distributivity hold:

$(F, +)$ is a commutative group	(F^*, \cdot) is a commutative group
$\forall_{a,b \in F} a + b = b + a \in F$	$\forall_{a,b \in F} a \cdot b = b \cdot a \in F$
$\exists_{0 \in F} \forall_{a \in F} a + 0 = 0 + a = a$	$\exists_{1 \in F^*} \forall_{a \in F^*} a \cdot 1 = 1 \cdot a = a$
$\forall_{a \in F} \exists_{-a \in F} a + (-a) = (-a) + a = 0$	$\forall_{a \in F^*} \exists_{a^{-1} \in F^*} a \cdot a^{-1} = a^{-1} \cdot a = 1$
$a \cdot (b + c) = a \cdot b + a \cdot c$	

(a) How do addition and multiplication have to be defined in $\mathbb{GF}(2) = \{0, 1\}$, the Galois⁸ field of order 2, i.e. with two elements?

⁸ Evariste Galois (1811-1832)

- (b) How are addition and multiplication to be defined in $\mathbb{GF}(3) = \{0, 1, 2\}$, the Galois field of order 3 ?
- (c) How are addition and multiplication to be defined in $\mathbb{GF}(5) = \{0, 1, 2, 3, 4\}$, the Galois field of order 5 ?
- (d) How can this approach be generalized to $\mathbb{GF}(p) = \{0, 1, 2, \dots, p-1\}$, the Galois field of prime order p ? Why is this approach doomed to failure for $\mathbb{GF}(pq)$ with primes p and q , i.e. for $\mathbb{GF}(m)$ with composite m ?

• **Galois Fields** $\mathbb{GF}(p^n)$

PROBLEM 22. Let p be prime and $n \in \mathbb{N}$. If $\mathbb{GF}(p^n)$ is defined to be a subset of $\mathbb{P}(n)$, the set of all polynomials of order n , i.e. of degree $n-1$, with coefficients in $\mathbb{GF}(p)$, so called polynomials *over* $\mathbb{GF}(p)$, then two such polynomials over $\mathbb{GF}(p)$ are readily as usual added.

- (a) What is then $(\mathbb{GF}(p^n), +)$?
- (b) What happens if two polynomials $r, s \in \mathbb{GF}(p^n)$ are multiplied as polynomials over $\mathbb{GF}(p)$?
- (c) Assuming the product of two polynomials $r, s \in \mathbb{GF}(p^n)$ is defined as the remainder of the product of r and s as polynomials over $\mathbb{GF}(p)$, divided by some polynomial m . How has such a polynomial m to look like, if each product so defined lies necessarily again in $\mathbb{GF}(p^n)$?
- (d) Which polynomials $m(x)$ have to be excluded in order to guarantee that products of non-vanishing factors do not vanish?
- (e) E.g., why is $m_1(x) = x^2 + 1$ a reducible and $m_2(x) = x^2 + x + 1$ an irreducible polynomial over $\mathbb{GF}(2)$?

PROBLEM 23.

- (a) How do multiplication and computation of inverse elements in $\mathbb{GF}(2^2)$ with $m(x) = x^2 + x + 1$ look like?
- (b) Let $m(x)$ be an irreducible polynomial over $\mathbb{GF}(p)$ of degree n . Defining a multiplication by

$$r \cdot s := (r(x) \cdot s(x)) \bmod m(x)$$

for $r, s \in \mathbb{GF}(p^n)^*$ then, what is $(\mathbb{GF}(p^n)^*, \cdot)$?

- (c) How are inverse elements in $\mathbb{GF}(p^n)$ computed?
- (d) How many irreducible polynomials over $\mathbb{GF}(p)$ there are of a given (small) degree?
- (e) In constructing $\mathbb{GF}(p^n)$, what impact has the choice of the irreducible polynomial $m(x)$ over $\mathbb{GF}(p)$ of degree $n - 1$?
- (f) Which elements generate e.g. $\mathbb{GF}(2^2)^*$ or $\mathbb{GF}(2^3)^*$?
- (g) How can the cyclicity of $\mathbb{GF}(p^n)^*$ be used to speed up the multiplication in $\mathbb{GF}(p^n)^*$?
- (h) How can the cyclicity of $\mathbb{GF}(p^n)^*$ be used to speed up the inversion in $\mathbb{GF}(p^n)^*$?

4. Cryptography

• Caesar and Cohorts

PROBLEM 24.

Let the letters of the Latin alphabet be numbered from 0 to 25 !

(a) Caesar⁹- encryption/decryption:

Plain text $x_1x_2x_3 \dots$ is letter-wise encrypted by key k per

$$y = (x + k) \bmod 26 \quad \text{to give the encrypted text } y_1y_2y_3 \dots$$

Encrypted text $y_1y_2y_3 \dots$ is letter-wise decrypted by key k per

$$x = (y - k) \bmod 26 \quad \text{to give the plain text } x_1x_2x_3 \dots$$

There is a encrypted text **wklvldwrsvhfuhwphvvdjh**.

(b) How many keys are there? What degree of security is achieved?

⁹Gaius Julius Caesar (100-44 v.Chr.)

- **Caesar in General**

PROBLEM 25.

Let the letters of the Latin alphabet be numbered from 0 to 25 !

- (a) Under which condition is $y = (kx) \bmod m$ a useful encryption method?
- (b) When encrypting per $y = (kx) \bmod m$ and decrypting per $x = (k^{\text{inv}}y) \bmod m$ what k^{inv} has to be used?
- (c) Combining both methods gives encryption per $y = (k_1 x + k_o) \bmod m$ and decryption per $x = (k'_1 y + k'_o) \bmod m$ using which k'_1 and k'_o ?
- (d) How many keys are there? What degree of security is achieved?

• Vigenère and Accomplices

PROBLEM 26.

Let the letters of the Latin alphabet be numbered from 0 to 25 !

(a) Vigenère¹⁰-encryption/decryption:

Plain text $x_1x_2x_3\dots$ is letter-wise encrypted to encrypted text $y_1y_2y_3\dots$ per $y_i = (x_i + k_i \bmod l) \bmod 26$ using key $k_0k_1\dots k_{l-1}$,
 encrypted text $y_1y_2y_3\dots$ is letter-wise decrypted to plain text $x_1x_2x_3\dots$ per $x_i = (y_i - k_i \bmod l) \bmod 26$ using key $k_0k_1\dots k_{l-1}$.
dlgcmqkxmzwcmvcdqccwyqi is an encrypted message.

(b) How many keys are there? What degree of security is achieved?

¹⁰ Blaise de Vigenère (1523-1596)

• Permutations

PROBLEM 27. For Caesar- and Vigenère-encryption/decryption it is characteristic that due to one (Caesar) or several (Vigenère) one-to-one functions $f : \mathcal{A} \rightarrow \mathcal{A}$ of the used alphabet \mathcal{A} each plain text letter is substituted by another (*monoalphabetic substitution*). Such functions f are also called *permutations*.

- (a) The Latin alphabet $\mathcal{A} = \{A, B, \dots, Z\}$ has 26 letters. How many permutations of \mathcal{A} there are?
- (b) Do permutations provide new encryption/decryption methods – essentially better than the Caesar- or the Vigenère-method?
- (c) How feasible is encryption by just permuting the plain text letters?

- DES

PROBLEM 28. *Data Encryption Standard, DES* [27] is a block oriented, symmetrical (identical keys for encryption and decryption) encryption/decryption method consisting of permutations and several substitutions, s.a. www.itl.nist.gov/fipspubs/fip46-2.htm

- (a) The DES algorithm applies an initial permutation P , then several substitutions, and finally P^{inv} to each 64bit block of the plain text. DES specifies P as follows

$$P = \begin{array}{cccccccc} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 & 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 & 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \\ 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 & 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 & 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{array} \quad \text{What is } P^{\text{inv}}?$$

- (b) Each of the other operations encrypts left and right 32bit halves of a 64bit block (L, R) by 32bit key K per

$$f_K(L, R) = (R, L \oplus K) \quad \text{where } \oplus \text{ denotes addition modulo 2}$$

$$f_K^{\text{inv}}(L, R) = ? \quad \text{In what respect are these operations substitutions?}$$

- (c) What type of encryption has been defined by $L := P^{\text{inv}} \circ f_{K_{16}} \circ f_{K_{15}} \circ \dots \circ f_{K_2} \circ f_{K_1} \circ P$ so far? with what consequences?
- (d) The last element in DES is a confusion/diffusion¹¹-method which is implemented by the so called *substitution boxes*, *S-Boxes*: each half block à 32bit is extended to 48bit by duplicating certain bits (depending on the round): a total of eight S-Boxes S_1, \dots, S_8 encrypt 6bit input to 4bit output each, e.g. S_5

S_5 Outer 2 bits	middle four bits of input															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0010	1100	0100	0001	0111	1100	1011	0110	1000	0101	0011	1111	1101	0000	1110	0001
01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1100	0011	1001	1000	0110
10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	0100	0101	0011

s.a. www.itl.nist.gov/fipspubs/fip46-2.htm or e.g. also www.kuno-kohn.de/crypto/crypto/des.htm for all eight DES S-boxes. How big are the look up tables for the eight S-boxes altogether? How big would the look up table for the 32bit substitution implemented by the S-boxes be? How to invert a S-box?

¹¹ Claude E. Shannon (1916-2001)

PROBLEM 29. Since its publication, the security of the *Data Encryption Standard*, *DES* was disputed, cp. e.g.

http://en.wikipedia.org/wiki/Data_Encryption_Standard.

In mid 1990ies, the insecurity of DES was demonstrated. This spurred improvements especially for high security critical applications.

- (a) What is effective DES key length? what is the DES key space?
- (b) *Triple DES*, *TDES* or *Triple Data Encryption Algorithm*, *TDEA* consists in applying DES three times with three keys

$$\text{TDES}_{K_3, K_2, K_1}(x) = \text{DES}_{K_3}(\text{DES}_{K_2}^{\text{inv}}(\text{DES}_{K_1}(x)))$$

What condition guaranties that several DES encryptions (like TDEA) offer substantially higher security?

- (c) What is effective TDEA key length? what is the TDES key space?
- (d) When and by what has DES resp. TDEA been superseded?

- **Public Keys?**

PROBLEM 30. Symmetric encryption/decryption methods require that the key (identical for encryption and decryption) can be exchanged between sender and receiver via a secure channel – a contradiction per se!

Asymmetric encryption/decryption methods working with pairs of private, i.e. secret and public key, so called *public key encryption methods*¹² do offer a solution.

(a) For each partner A, B, C, \dots there is a public key and hence a public encryption method f_A, f_B, f_C, \dots . Each partner keeps her/his private key $A^{-1}, B^{-1}, C^{-1}, \dots$ and hence her/his private decryption method $f_A^{-1}, f_B^{-1}, f_C^{-1}, \dots$ top secret.

Now, Bob can tell Alice say x by sending to her the encrypted message $f_A(x)$. Only Alice can decrypt this message by f_A^{-1} to get $x = f_A^{-1}(f_A(x))$.

What is the base of the security of such *public key* methods?

¹² Whitfield Diffie, Martin Hellman: New Directions in Cryptography; IEEE Trans. Inform. Theory, IT-22, 6, Nov 1976 pp.644-654

- **RSA**

PROBLEM 31. The RSA¹³-method is a public key encryption/decryption method. It works as follows:

Let p and q be big prime numbers and $n = p \cdot q$, i.e. $\varphi(n) = (p-1)(q-1)$.

A message x is *encrypted* by

$$\boxed{y = x^e \bmod n} \text{ with public key } e, \text{ so that } \gcd(e, \varphi(n)) = 1.$$

A message y is (*decrypted*) by

$$\boxed{x = y^d \bmod n} \text{ with private key } d, \text{ so that } ed = 1 \bmod \varphi(n).$$

- (a) Show: $f_e : x \rightarrow x^e \bmod n$ is a trapdoor function.
- (b) The security of the RSA-method, on what basis does it rest?
- (c) f_e^{-1} , i.e. f_d can be used to generate a digital signature.

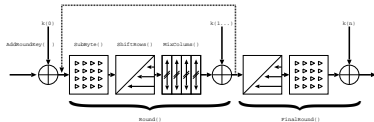
If Alice signs her message digitally, then Bob is assured that a message y he received truly originated by Alice. How to cut cost?

¹³R. Rivest, A. Shamir, L. Adleman: A method for obtaining digital signatures and public key cryptosystems; Communications ACM, 21 (1978), 120-126

- AES

PROBLEM 32. Established in 2000, the *Advanced Encryption Standard*, *AES* is DES's successor standard. To avoid all suspicions of conspiracy of the standardizing body (NIST) with the developers of the standard (IBM in the case of DES) this standard is the result of a public competition. AES represents a special case of the Rijndael cipher [25].

- What type of cipher is AES?
- What are the characteristic parameters of AES.
- What does a round of AES consist of?



PROBLEM 33. Now, the functions of a round are to be examined separately. Identifiers are used as in

csrc.nist.gov/publications/fips/fips197/fips-197.pdf

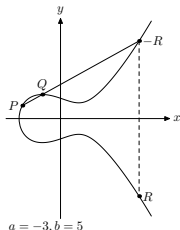
- (a) **SubBytes()**: How is this substitution specified? How is it implemented by a s-box? How is the substitution inverted?
- (b) **ShiftRows()**: How is the permutation of the rows of a block implemented when a block is represented as 4×4 -byte-matrix? How is this transformation inverted?
- (c) **MixColumns()**: How are the columns of a block transformed when a block is represented as 4×4 -byte-matrix? How is this transformation inverted?
- (d) **AddRoundKey()**: How are the columns of a block XORed by parts of the expanded key? Why is this transformation its own inverse?

- **Elliptic Curves over \mathbb{R}**

PROBLEM 34. To introduce *Elliptic Curve Cryptography*, *ECC* it is reasonable to consider so called *elliptic curves* $y^2 = x^3 + ax + b$ over \mathbb{R} , i.e. curves in \mathbb{R}^2 with real coefficients $a, b \in \mathbb{R}$ first.

- (a) Which geometric features exhibit elliptic curves $E = E(\mathbb{R}) = E_{a,b}(\mathbb{R})$ over \mathbb{R} ? What happens for $x \rightarrow +\infty$?
- (b) What are the zeroes of the radicand $x^3 + ax + b$ of an elliptic curve $E = E(\mathbb{R}) = E_{a,b}(\mathbb{R})$ over \mathbb{R} ?
- (c) What condition guaranties that the radicand of an elliptic curve $E = E(\mathbb{R}) = E_{a,b}(\mathbb{R})$ over \mathbb{R} has no multiple zeroes?
- (d) Given any non vertical, non tangent line intersecting an elliptic curve $E = E(\mathbb{R}) = E_{a,b}(\mathbb{R})$ over \mathbb{R} at least twice. Why does the line then intersect the curve E trice?
- (e) Given $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ with $x_P \neq x_Q$ on an elliptic curve $E = E_{a,b}(\mathbb{R})$ over \mathbb{R} . Under the above condition, what are the coordinates of the third intersection point $R = (x_R, y_R)$ on E and on the line through P and Q ?

PROBLEM 35. One specifies an addition of points P and Q on an elliptic curve $E = E_{a,b}(\mathbb{R})$ over \mathbb{R} by defining $R := P + Q$ to be the the intersection point of the line through P and Q with E , mirrored at the x -axis.



- How is $P + P$ to be defined consistently?
- How can $P + Q$ for $x_P = x_Q$ and $y_P \neq y_Q$ be defined consistently?
- What does this mean for $P + Q$ with $P = (x_P, y_P)$ and $Q = (x_P, -y_P)$ and the solvability of $P + Q = R$ in Q for given $P, R \in E$?
- Which structure on $E = E_{a,b}(\mathbb{R})$ is provided by this addition?

- **Elliptic Curves over $\mathbb{GF}(p)$**

PROBLEM 36. Elliptic curves $E = E_{a,b}(\mathbb{R})$ over \mathbb{R} are unfit for cryptographic purposes. Instead one uses elliptic curves $E = E_{a,b}(\mathbb{F})$ over some finite field \mathbb{F} , e.g. $\mathbb{F} = \mathbb{GF}(p)$ for prime p .

(a) How is $P + Q$ to be defined on $E = E_{a,b}(\mathbb{GF}(p))$?

- **Elliptic Curves over $\mathbb{GF}(2^m)$**

PROBLEM 37. Using $\mathbb{F} = \mathbb{GF}(2^m)$, another type of finite fields, allows to define groups on elliptic curves $E = E_{a,b}(\mathbb{GF}(2^m))$ over $\mathbb{GF}(2^m)$, namely

$$y^2 + xy = x^3 + ax + b \quad \text{for } a, b \in \mathbb{GF}(2^m)$$

(a) Why can now $y^2 = x^3 + ax + b$ be used no longer?

(b) How is $P + Q$ to be defined on $E = E_{a,b}(\mathbb{GF}(2^m))$?

- **Elliptic Curve Cryptography, ECC**

PROBLEM 38. Elliptic Curve Cryptography, ECC is based on exploiting the group structure of a public elliptic curve $E = E_{a,b}(\mathbb{F})$ over some finite field \mathbb{F} together with some suitable generator point $G \in E$. Each participant owns a secret and public key pair $(r, Q) \in \mathbb{N} \times E$ with random number $1 < r < \text{card}(\langle G \rangle)$ and $Q = rG$.

- (a) What type of cipher is ECC, suitable for what applications?
- (b) How can an ECC based El-Gamal encryption/decryption be implemented?
- (c) How can an ECC based Diffie-Hellman key exchange, *ECDH*, be implemented?
- (d) How can an ECC based Digital Signature Algorithm, *ECDSA* be implemented?

5. Compression

• Exploiting Relative Frequencies

PROBLEM 39.

If the (relative) frequencies of the symbols in a text are known a priori then one can design a code so that the most frequent symbols are assigned the shortest codes. Let us call such codings *monotonous*. To save the insertion of a special character to separate codes it is necessary that each code cannot be confused with the beginning of another code: The coding has to be *prefix-* or *comma-free*.

- (a) Given an alphabet s_1, s_2, \dots, s_n with frequency f_i of symbol s_i , where $f_1 > f_2 > \dots > f_n$ for $i = 1, \dots, n$. Assume $c_i = \text{code}(s_i) = 01^{i-1} \in \{0, 1\}^i$. What about this code?
- (b) How to represent prefix-free codings by graphs?
- (c) Construct a monotonous prefix-free coding.

- **Using Dictionaries**

PROBLEM 40.

The idea of LZW¹⁴ is to let sender and receiver set up and maintain a dictionary for characters and combination of characters to be sent and received.

(a) Both in compression and decompression, first the dictionary is initialized with the letters of the alphabet together with their codes. Then, plain text resp. compressed text is read character by character.

In compression, the text is read character by character. **PATTERN** is the longest string in the dictionary which coincides with the recently read input characters. In decompression the codes are read. At the same time, the dictionary is accordingly extended.

¹⁴ Jacob Ziv and Abraham Lempel: A Universal Algorithm for Sequential Data Compression; IEEE Transactions on Information Theory, May 1977
Terry Welch, "A Technique for High-Performance Data Compression", Computer, June 1984

Compression:

```
PATTERN = get input character
WHILE there are still input characters DO
    CHARACTER = get input character
    IF PATTERN+CHARACTER is in dictionary
        PATTERN = PATTERN+character
    ELSE
        output the code for PATTERN
        add PATTERN+CHARACTER to dictionary
        PATTERN = CHARACTER
output the code for PATTERN
```

Decompression:

```
Read oldCODE; output dict[oldCODE]
WHILE there are still input characters DO
    Read newCODE
    PATTERN = dict[newCODE]
    output PATTERN
    CHARACTER = first character in PATTERN
    add dict[oldCODE]+CHARACTER to dictionary
    oldCODE = newCODE
```

(b) There is a flaw in the algorithm presented above:

6. Probability & Intuition

- **Cards & Goats**

PROBLEM 41.

- (a) In an urn there are three cards: one is on both sides red, one on both sides blue, and the third one is on one side red and on one side blue.

What is the probability P that a card drawn at random from the urn is red on the top side and blue on the bottom side?

- (b) In a contest there are three doors behind which two goats and a car are hidden (the quizmaster knows where).

The candidate chooses a door. Then the quizmaster reveals a goat behind another door.

Does the candidate improve the chances to win the car by revising her/his initial choice?

• Algorithms to Generate Chance?

Random numbers play an important role in simulation, (zero knowledge) authentication etc. Hence, high level programming languages usually offer library functions like `ran`, `random` or `randomize` to algorithmically and hence deterministically generate so called *pseudo-random numbers*.

PROBLEM 42.

- (a) What are characteristics of random numbers besides being seemingly random (whatever this might be)? How to generate random numbers with such given characteristics from random numbers of some standard?
- (b) How to generate standard random numbers fast, i.e. by little computational effort?
- (c)
$$x_{n+1} = (a x_n + c) \bmod m, \quad \text{mit } x_0 = 1$$
 is periodic – why? and with which maximal/minimal periodic length?

- **What is Randomness?**

Criteria for the quality of pseudo random number generators have to be established, especially of generators of evenly distributed, continuous pseudo random numbers in the unit interval. These criteria are to be assessed in tests.

But, randomness has no definition, no specification. Therefore, there can be tests only for certain features of randomness.

PROBLEM 43.

- (a) How to test whether the co-domain is evenly covered?
- (b) How to test randomness of pseudo random numbers by measuring the information content of each generated digit?
- (c) How to test randomness of pseudo random numbers by measuring their compressability?
- (d) How to test randomness of pseudo random numbers by measuring the mutual (in) dependence of their digits?

7. Sources and Links

Some references on Recreational Mathematics

- [1] About.com: Recreational Mathematics;
<http://math.about.com/od/recreationalmath>
- [2] Bild der Wissenschaft; www.wissenschaft.de/ s. Spiele-Archiv
- [3] Chlond, Martin: Integer Programming in Recreational Mathematics;
www.chlond.demon.co.uk/academic/puzzles.html
- [4] Canadian Mathematical Society www.math.ca/Recreation
- [5] Dutch, Steven: Recreational Mathematics;
www.uwgb.edu/dutchs/RECMATH/recmath.htm
- [6] Eppstein, David: Math Fun;
www.ics.uci.edu/~eppstein/recmath.html
- [7] Flannery, Sarah: In Code – A Mathematical Journey; Profile Books, 2000 ISBN 1-86197-222-9 [2](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#)

- [8] Gardner, Martin: *Mathematical recreations* and many more titles; s. book list, e.g. <http://thinks.com/books/gardner.htm>
- [9] Gilleland, Michael: Recreational Mathematics Links;
www.merriampark.com/maths.htm
- [10] Google Directory - Science > Math > Recreations;
www.google.com/Top/Science/Math/Recreations
- [11] Journal of Recreational Mathematics, Editor: Charles Ashbacher and Lamarr Widmer;
www.baywood.com/journals/PreviewJournals.asp?Id=0022-412x
- [12] Mathematical Association of America, MAA: Recreational Mathematics;
www.maa.org/BLL/recmath.htm
- [13] Mathematikwettbewerb Känguru e.V. www.mathe-kaenguru.de
s.a. www.weblearn.hs-bremen.de/risse/MAI/docs/
- [14] Mitchon, Gerald P.: Recreational Mathematics;
<http://home.att.net/~numericana/answer/recreational.htm>
- [15] New Scientist? www.newscientist.com

- [16] O'Connor, J.J., Robertson, E.F.: mathematical games and recreations;
www-groups.dcs.st-andrews.ac.uk/~history/HistTopics/Mathematical_games.html
- [17] open directory project dmoz.org/Science/Math/Recreations/
- [18] Problem of the Week, s.
www.google.de/search?...&q=problem+of+the+week ...
- [19] Scientific American www.sciam.com,
s. *puzzling adventures* in single issues
- [20] Singmaster, David: The Unreasonable Utility of Recreational Mathematics;
anduin.eldar.org/~problemi/singmast/ecmutil.html
- [21] Eugène Strens Recreational Mathematics Collection Database;
www.ucalgary.ca/lib-old/sfgate/strens
- [22] Wilkinson, David: Recreational Mathematics Links;
www.scit.wlv.ac.uk/~cm1985/RecMaths.html
- [23] Wolfram Mathworld: Recreational Mathematics;
mathworld.wolfram.com/topics/RecreationalMathematics.html

Some references on Number Theory

- [24] Forster, Otto: Algorithmische Zahlentheorie; Vieweg 1996 93, 100

Some references on Cryptography

- [25] Federal Information Processing Standards, FIPS: Advanced Encryption Standard (AES); Publication 197
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
Advanced Encryption Standard Algorithm Validation List
<http://csrc.nist.gov/cryptval/aes/aesval.html> 36
- [26] Daemen, Joan, Rijmen, Vincent: The Design of Rijndael – AES, The Advanced Encryption Standard; Springer 2002
- [27] Federal Information Processing Standards, FIPS: Data Encryption Standard (DES); Publication 46-3 <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf> 31

- [28] Federal Information Processing Standards, FIPS: Digital Signature Standard (DSS) – DSA, RSA, and ECDSA algorithms; Publication 186-2 <http://csrc.nist.gov/cryptval/dss.htm>
- [29] Hankerson, Darrel, Menezes, Alfred, Vanstone, Scott: Guide to Elliptic Curve Cryptography; Springer 2004 170
- [30] Oswald, Elisabeth: Introduction to Elliptic Curve Cryptography; www.iaik.tugraz.at/aboutus/people/oswald/papers/Introduction_to_ECC.pdf 164, 167, 170
- [31] Standards for Efficient Cryptography Group, SECG: SEC1 – Elliptic Curve Cryptography; www.secg.org/collateral/sec1_final.pdf 170
- [32] Wagner, Neal R.: The Laws of Cryptography; www.cs.utsa.edu/~wagner/lawsbookcolor/laws.pdf

Some references on Coding Theory and Compression

- [33] Dankmeier, Wilfried: Codierung; Vieweg 2001

- [34] Nelson, Marc, Gailly, Jean-loup: The Data Compression Book; 2nd edition, M&T Books, New York, NY 1995

Some references on Probability

- [35] Bronstein, I.N. & Semendjajew, K.A. et al: (Teubner-) Taschenbuch der Mathematik; Teubner 2003 190

Of course, any feedback, critics, inventive problems and solutions are most welcome.

Prof. Dr. Th. Risse, ZIMT 244,

www.weblearn.hs-bremen.de/risse

0049 (0)421 5905-5489

[mailto: risse@hs-bremen.de](mailto:risse@hs-bremen.de)

Solutions to Problems

Problem 1(a)

$$4 = (5 - 3) + (5 - 3)$$



Problem 1(b)

$$1 = 3 + 3 - 5$$



Problem 1(c)

For example, $5 = 9 - 4$, $3 = 4 + 4 + 4 - 9$, ...



Problem 1(d)

There is **no** solution.



Problem 2(a)

This is the height he achieves each day:

At the 1st day he reaches 300m,

at the 2nd day he reaches 400m,

at the 3rd day he reaches 500m, ...

at the 18th day he reaches 3000m



Problem 2(b)

A needed t_A time units, TU for the 100m. Hence his speed is $v_A = 100/t_A$.

B needed t_B TU for the 100m. Hence his speed is $v_B = 100/t_B = 90/t_A$. Therefore $t_A/t_B = 0.9$.

The speed of C is $v_C = 90/t_B = x/t_A$. Therefore $x = 90t_A/t_B = 90 \cdot 0.9 = 81$ m.

Thus, the first runner A beats C by 19m. □

Problem 3(a)

$36 = 2^2 \cdot 3^2$. If one considers also the one year olds, then there are the following combinations:

3.	2.	1.	Σ
1	1	36	38
1	2	18	21
1	3	12	16
1	4	9	14
1	6	6	13
2	2	9	13
2	3	6	11
3	3	4	10

Only in case of sum 13 another hint was necessary. But there is an oldest child only if the family has two years old twins and a six years old child. □

Problem 4(a) poor man's solution:

The catastrophe happens after one hour. Then, the fly has travelled 75km.

allegedly **John von Neumann's** solution:

Let s_l be the position of the 'left' train, s_r that one of the 'right' train. The fly started 'right'. Let t_1 be the point in time when the fly meets the 'left' train, t_2 when it meets the 'right' trains, etc.

Then, we have $s_l(t_1) = 50t_1$ and $75 = (100 - s_l(t_1))/t_1$. Hence, $75t_1 = 100 - 50t_1$, $125t_1 = 100$ and finally $t_1 = 4/5$.

t/h	$s_l(t)/km$	$s_r(t)/km$	d/km
0	0	100	0
4/5	40	60	60
4/5 + 4/25	48	52	12
4/5 + 4/25 + 4/125	49.6	50.4	2.4
	\vdots		

where d is the distance the fly has travelled between two impinge-

ments.

The point in time of the catastrophe is

$$t_{\infty} = 4 \sum_{i=1}^{\infty} 0.2^i = 4 \left(\frac{1}{1-0.2} - 1 \right) = 4 \left(\frac{5}{4} - 1 \right) = 1$$

and the total travelled distance

$$75 \frac{4}{5} + 75 \frac{4}{25} + 75 \frac{4}{125} \dots = 75 \text{km.}$$



Problem 5(a)

Let $s_h(t)$ and $s_r(t)$ denote the distance travelled along the outward journey and the return journey resp., at any point t in time between sunrise 0 and sunset 1.

Let d denote the total distance between A and B . Then, $s_h(0) = 0$, $s_h(1) = d$, $s_r(0) = d$, $s_r(1) = 0$.

With s_h and s_r also $\delta(t) = s_r(t) - s_h(t)$ is a continuous function of t . Because of the different signs of $\delta(0) = d$ and $\delta(1) = -d$ the function $\delta(t)$ has at least one zero t_o in the intervall $[0, 1]$.

At time t_o we have $s_r(t_o) = s_h(t_o)$.

Under which conditions are there more than one such point? □

Problem 6(a) There are eight equations with nine unknowns. And, the solutions have to consist of the natural numbers $1, \dots, 9$.

Stepwise pick the arrangements corresponding to magic squares from a total of $9! = 362880$ arrangements.

1. The number in the middle/centre is necessarily 5.

It cannot be $n = 6, 7, 8$ or 9 because then $m = 9, 8, 7$ or 6 had no place in the magic square.

2. The 9 is in no corner, neither in NO, NW, SW, nor SO.

Assuming NW=9 the SO=1 and for the **three** numbers 6,7 and 8 there would be left only the **two** positions O and S.

3. Without restriction of generality let W=9, then either NW=2 and SW=4 or NW=4 and SW=2.

Assuming now NW=3. Then also SW=3. But the number 3 must not appear twice.

Two of the eight possible magic squares – identical when taking symmetry into consideration – are presented

2	7	6
9	5	1
4	3	8

und

4	3	8
9	5	1
2	7	6

How do the other six magic squares look like?



Problem 7(a)

This text has **no** letter **e**, but every other letter of the Latin alphabet occurs at least once.

Write a similar text in German.



Problem 8(a)

With the following procedure he acquires the average salary without him or any of his employees getting to know an individual salary.

1. He at random chooses a big 'secret' number k .
2. He tells k to the first employee in order to increment k by her/his own salary and to tell the sum to the second employee.
3. One after another the employees get to know some number in order to increment it by their own salary and to tell the sum to the next colleague.
4. The last n^{th} employee increments the number by her/his salary and tells the sum g to the boss.

Then, the average salary is $(g - k)/n$.



Problem 9(a)

The problem is to let Alice get at the content of the box. The two agree on the following procedure:

1. Bob sends the box locked by his lock to Alice.
2. Alice additionally locks the box she received by her lock and sends it back to Bob.
3. Bob removes his lock from the box and sends the box locked by only Alice's lock back to Alice.



Problem 10(a)

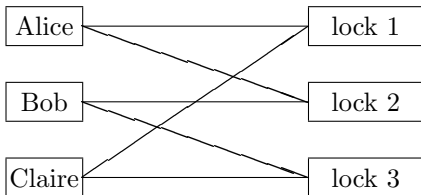
They agree – say per e-mail – on the following procedure:

1. Alice and Bob agree to use a suitable *one way function* f , i.e. a one-to-one function $f : \mathbb{N} \supset D \rightarrow W \subset \mathbb{N}$, so that $f(x)$ is easily and $f^{\text{inv}}(y)$ is extremely hard to compute.
2. Now, say Alice starts and chooses an odd or even $x \in D$. Now she sends y with $y = f(x)$ to Bob without offenbaren x .
3. Bob receives y and bets whether x was odd or even.
(*If he wins then Alice otherwise Bob has to drive.*)
4. Alice checks Bob's bet and sends x to Bob for verification, i.e. to let Bob compare $f(x)$ with the y he initially received.



Problem 11(a)

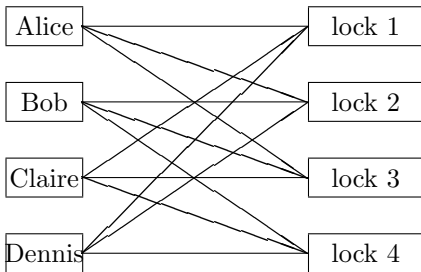
They need **three** locks with **two** keys each. If each person owns keys according to the following scheme,



then only at least two persons together have keys for all three locks of the treasure box. □

Problem 11(b)

The following schema represents a solution



because one person is always lacking a key, and any two persons together have a key to each of the four locks.

Is this solution with **four** locks with **three** keys each minimal?

With three locks each person may have at most two keys. Hence there are a total of at most eight keys for three locks and for four persons:

There is no lock with only one key, because without the owner of that one key pairs of persons cannot open the treasure box.

Hence there is either one person with keys to four locks or there are two persons with keys to three locks. In both cases a contradiction!

Finally with four locks, it is not sufficient to have two keys per person because then any two persons together might not have keys to each of the four locks! □

Problem 11(c) ?



Problem 12(a) The first three Fermat numbers

$F(1) = 2^{2^1} + 1 = 5$, $F(2) = 2^{2^2} + 1 = 17$, $F(3) = 2^{2^3} + 1 = 257$
can easily be verified to be prime. Using `calc.exe`, a pocket calculator, www.weblearn.hs-bremen.de/risse/MAI/docs/numerics.pdf, etc. also the fourth Fermat number $F(4)$ is verified to be prime.

$$F(4) = 2^{2^4} + 1 = 65537$$

Not until Euler¹⁵ it was achieved to factorise the fifth Fermat number

$$F(5) = 2^{2^5} + 1 = 4294967297 = 641 \cdot 6700417$$

This and a fortiori factorisation of the sixth Fermat number

$$F(6) = 2^{2^6} + 1 = 18446744073709551617 = 274177 \cdot 67280421310721$$

today is conveniently possible using powerful tools like Mathematica, Maple, MATLAB, MuPAD, etc. (cp. [/risse/symbolic/](#)) □

¹⁵ Leonhard Euler (1707-1783)

Problem 13(a)

For example,

$$E(41) = 1681 = 41^2$$

and similarly

$$E(42) = 1763 = 41 \cdot 43.$$

To carry on s.a. www.weblearn.hs-bremen.de/risse/MAI/,
www.cs.unb.ca/profs/alopez-o/math-faq/math-faq.pdf



Problem 14(a)

Namely, $M(11) = 2047 = 23 \cdot 89$.

The Lucas¹⁶-Lehmer¹⁷-test, s. e.g. (3.2.8 *What is the current status on Mersenne primes?*) of

www.cs.unb.ca/profs/alopez-o/math-faq/math-faq.pdf tests efficiently whether a Mersenne number is prime or not. 1999 a record in the *Great Internet Mersenne Prime Search (GIMPS)*, was established showing that $M(6972593)$ – a number with 2098960 digits – is prime.

Everybody can provide idle cycles of PC's to compute prime Mersenne numbers, s. *Great Internet Mersenne Prime Search (GIMPS)*

GIMPS runs many more projects of *distributed computing*. □

¹⁶François E.A. Lucas (1842-1891)

www-history.mcs.st-and.ac.uk/Biographies/Lucas.html

¹⁷Derrick N. Lehmer (1867-1938)

www.math.berkeley.edu/publications/newsletter/2000/lehmer.html

Problem 15(a)

It is the $(n \bmod 7)^{th}$ day of the week if we arrange the days of the week cyclically numbered from 0 to 6 starting with today's day of the week. □

Problem 15(b)

It is the $((7 - (n \bmod 7)) \bmod 7)^{th}$ day of the week if we arrange the seven days of the week cyclically numbered from 0 to 6 starting with today's day of the week. \square

Problem 15(c)

E.g. see

www.cl.cam.ac.uk/~mgk25/iso-time.html



Problem 16(a)

Per definition we have for $m, n, r \in \mathbb{N}$

$$\begin{aligned}n \bmod m = r &\iff n = v \cdot m + r \text{ für ein } v \in \mathbb{N} \\ &\iff n - r = v \cdot m \text{ für ein } v \in \mathbb{N} \\ &\iff m \mid n - r \iff n \equiv r \pmod{m}\end{aligned}$$



Problem 16(b)**additivity:**

$$\begin{array}{rcl}
 n_1 \equiv r_1 \pmod{m} & \Rightarrow & n_1 - r_1 = v_1 m \\
 n_2 \equiv r_2 \pmod{m} & \Rightarrow & n_2 - r_2 = v_2 m \\
 \hline
 (n_1 \pm n_2) \equiv (r_1 \pm r_2) \pmod{m} & \Leftarrow & n_1 + n_2 - (r_1 + r_2) = (v_1 + v_2)m
 \end{array}$$

multiplicativity:

$$\begin{aligned}
 n_i \equiv r_i \pmod{m} &\Rightarrow \left\{ \begin{array}{l} m | n_1 - r_1 \\ m | n_2 - r_2 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} m | r_2(n_1 - r_1) \\ m | n_1(n_2 - r_2) \end{array} \right\} \Rightarrow \\
 m | r_2(n_1 - r_1) + n_1(n_2 - r_2) &= n_1 n_2 - n_1 r_2 + n_1 r_2 - r_1 r_2 \\
 &\Rightarrow n_1 \cdot n_2 \equiv r_1 \cdot r_2 \pmod{m}
 \end{aligned}$$



Problem 16(c)**scalar multiples**

$$n \equiv r \pmod{m} \Rightarrow m|n - r \Rightarrow m|c(n - r) \Rightarrow c \cdot n \equiv c \cdot r \pmod{m}$$

powers either by multiplicativity or directly by induction: $p=1$ ✓

which leaves us to show $n^p \equiv r^p \pmod{m} \Rightarrow n^{p+1} \equiv r^{p+1} \pmod{m}$

$$\left. \begin{array}{l} n \equiv r \pmod{m} \\ n^p \equiv r^p \pmod{m} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} m|n - r \\ m|n^p - r^p \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} m|r^p(n - r) \\ m|n(n^p - r^p) \end{array} \right\} \Rightarrow$$

$$m|r^p(n - r) + n(n^p - r^p) = n^{p+1} - nr^p + nr^p - r^{p+1}$$

$$\Rightarrow n^{p+1} \equiv r^{p+1} \pmod{m}$$



Problem 16(d)**transitivity**

$$\begin{aligned}r \equiv s \pmod{m}, s \equiv t \pmod{m} &\Rightarrow m \mid r - s, m \mid s - t \\ &\Rightarrow m \mid (r - s) + (s - t) = r - t \Rightarrow r \equiv t \pmod{m}\end{aligned}$$



Problem 17(a)

Division by 3: (due to exponentiation)

$$10^0 = 1 \equiv 1 \pmod{3} \Rightarrow 10^p \equiv 1 \pmod{3}$$

and (due to multiplicativity)

$$z_i 10^i = z_i \equiv 1 \pmod{3} \Rightarrow n = \sum_{i=0}^{\infty} z_i 10^i \equiv \sum_{i=0}^{\infty} z_i \pmod{3}$$

Specially we have

$$s(n) \equiv 0 \pmod{3} \Rightarrow n \equiv 0 \pmod{3}$$

Division by 9: analogously! e.g.

$$1234567890 \pmod{3} = (1 + 2 + \dots + 9) \pmod{3} = 45 \pmod{3} = 0$$

$$1234567890 \pmod{9} = (1 + 2 + \dots + 9) \pmod{9} = 45 \pmod{9} = 0$$

The common tests for divisibility by 2,4 or 5 are deduced correspondingly.



Problem 17(b)

Remainders when dividing powers of 10 by 11:

$$\left. \begin{array}{l} 10^0 \equiv 1 \pmod{11} \\ 10^1 \equiv 10 \pmod{11} \end{array} \right\} \Rightarrow \begin{cases} 10^{2i} \equiv 1 \pmod{11} \\ 10^{2i+1} \equiv 10 \pmod{11} \equiv -1 \pmod{11} \end{cases}$$

Arithmetic modulo 11 gives

$$z_{2i}10^{2i} \equiv z_{2i} \pmod{11} \text{ and } z_{2i+1}10^{2i+1} \equiv -z_{2i+1} \pmod{11}$$

Together with transitivity we get a test for the divisibility by 11:

$$11 \mid \sum_{i=0}^{\infty} (-1)^i z_i \Rightarrow 11 \mid \sum_{i=0}^{\infty} z_i 10^i$$

and e.g.

$$1234567890 \pmod{11} = (-1 + 2 - 3 + 4 - 5 + 6 - 7 + 8 - 9 + 0) \pmod{11} = -5 \pmod{11} = 6 \quad \square$$

Problem 17(c)

Some examples may illustrate the procedure:

Check digit of ISBNNumber 1-86197-222 is 1-86197-222-9 because

$$1 \cdot 1 + 2 \cdot 8 + 3 \cdot 6 + 4 \cdot 1 + 5 \cdot 9 + 6 \cdot 7 + 7 \cdot 2 + 8 \cdot 2 + 9 \cdot 2 = 174 \pmod{11} = 9 \pmod{11}$$

Check digit of ISBNNumber 3-933146-67 is 3-933146-67-4 because

$$1 \cdot 3 + 2 \cdot 9 + 3 \cdot 3 + 4 \cdot 3 + 5 \cdot 1 + 6 \cdot 4 + 7 \cdot 6 + 8 \cdot 6 + 9 \cdot 7 = 48 \pmod{11} = 4 \pmod{11}$$

Check digit of ISBNNumber 3-933146-43 is 3-933146-43-7 because

$$1 \cdot 3 + 2 \cdot 9 + 3 \cdot 3 + 4 \cdot 3 + 5 \cdot 1 + 6 \cdot 4 + 7 \cdot 6 + 8 \cdot 4 + 9 \cdot 3 = 51 \pmod{11} = 7 \pmod{11}$$

Check digit of ISBNNumber 0-550-10206 is 0-550-10206-X because

$$1 \cdot 0 + 2 \cdot 5 + 3 \cdot 5 + 4 \cdot 0 + 5 \cdot 1 + 6 \cdot 0 + 7 \cdot 2 + 8 \cdot 0 + 9 \cdot 6 = 32 \pmod{11} = 10 \pmod{11}$$



Problem 17(d) Remainders when dividing powers of 10 by 7:

$$\left. \begin{array}{l} 10^0 \equiv 1 \pmod{7} \\ 10^1 \equiv 3 \pmod{7} \\ 10^2 \equiv 2 \pmod{7} \\ 10^3 \equiv 6 \pmod{7} \\ 10^4 \equiv 4 \pmod{7} \\ 10^5 \equiv 5 \pmod{7} \\ 10^6 \equiv 1 \pmod{7} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} 10^{7i+0} \equiv 1 \pmod{7} \equiv -6 \pmod{7} \\ 10^{7i+1} \equiv 3 \pmod{7} \equiv -4 \pmod{7} \\ 10^{7i+2} \equiv 2 \pmod{7} \equiv -5 \pmod{7} \\ 10^{7i+3} \equiv 6 \pmod{7} \equiv -1 \pmod{7} \\ 10^{7i+4} \equiv 4 \pmod{7} \equiv -3 \pmod{7} \\ 10^{7i+5} \equiv 5 \pmod{7} \equiv -2 \pmod{7} \\ 10^{7i+6} \equiv 1 \pmod{7} \equiv -6 \pmod{7} \end{array} \right.$$

Arithmetic modulo 7 implies for each $n = \sum_{i=0}^{\infty} z_i 10^i$

$$n \equiv \sum_{i=0}^{\infty} (z_{7i+0} + 3z_{7i+1} + 2z_{7i+2} - z_{7i+3} - 3z_{7i+4} - 2z_{7i+5} + z_{7i+6}) \pmod{7}$$

Together with transitivity we get a test for the divisibility by 7:

$$\begin{aligned} 7 \mid \sum_{i=0}^{\infty} (z_{7i+0} + 3z_{7i+1} + 2z_{7i+2} - z_{7i+3} - 3z_{7i+4} - 2z_{7i+5} + z_{7i+6}) \\ \Rightarrow 7 \mid \sum_{i=0}^{\infty} z_i 10^i \end{aligned}$$

and e.g. $1234567890 \pmod{7} = (2 \cdot 1 + 3 \cdot 2 + 1 \cdot 3 + 1 \cdot 4 - 2 \cdot 5 - 3 \cdot 6 - 1 \cdot 7 + 2 \cdot 8 + 3 \cdot 9 + 1 \cdot 0) \pmod{7} = (2 + 6 + 3 + 4 - 10 - 18 - 7 + 16 + 27) \pmod{7} = 23 \pmod{7} = 2$ □

Problem 17(e) *Parity* or *Cyclic Redundancy Check*, *CRC* are examples of Error Detecting Codes, EDC or even Error Correcting Codes, ECC. Using them hardenes data against corruption and loss of data when transmitting (LAN, wLAN, satellite, ...) or storing (HD, RAID, CD-ROM, DVD ...).

e.g. Set the parity bit b_o for odd or even parity such that the number of set bits in a bit string $b_1 \dots b_n$ inclusive parity bit b_o is odd or even resp. By a single parity bit single, i.e. 1-bit errors are detected.

$$\sum_{i=0}^n b_i = \begin{cases} 1 \bmod 2 & \text{für odd parity} \\ 0 \bmod 2 & \text{für even parity} \end{cases}$$

By the way, odd parity is standard¹⁸ for synchrone, even parity for asynchrone transmission.

e.g. Obviously it is more demanding to correct errors than only to detect errors. Correspondingly, algorithms to correct errors like CRC or Reed-Solomon-Codes are more complex.

¹⁸s. z.B. www.its.bldrdoc.gov/projects/t1glossary2000/_parity-check.html

Explanations to relevant procedures to correct errors can be found e.g. at

Cyclic Redundancy Codes, CRC, s.

<ftp.informatik.uni-trier.de/pub/Users-CTVD/sack/ep/CRC.txt>

Reed¹⁹-Solomon¹⁹-Code, s. www.4i2i.com/reed_solomon_codes.htm,
www.cs.cornell.edu/Courses/cs722/2000sp/ReedSolomon.pdf

...



¹⁹Irving Reed (1923-?), Gustave Solomon (1931-1996)

hotwired.lycos.com/synapse/feature/97/29/silberman2a_1.html

Problem 18(a)

Let $a < b$ (otherwise there is nothing to do).

Let $b = va + r$ with $r = b \bmod a$.

If $r = 0$ then $\gcd(a, b) = a$ and $\gcd(a, (va) \bmod a) = \gcd(a, 0) = a$.

If $r > 0$ then for $d = \gcd(a, r)$ we have $d \mid a$ and $d \mid r$ and therefore also $d \mid b = va + r$.

It remains to show that d is greatest divisor of a and b . For a $d' \in \mathbb{N}$ with $d' \mid a$ and $d' \mid b = va + r$ it follows $d' \mid r$. Due to $d = \gcd(a, r)$ we have $d' \mid d$.

In total $\gcd(a, r) = d = \gcd(a, b)$ is deduced.



Problem 18(c)

Proof of FLT see e.g. [24] S.54-55

If p is prime then $G = (\mathbb{Z}/p\mathbb{Z})^* = \{1, 2, \dots, p-1\}$ is a multiplicative group with $p-1$ elements (i.e. closed under multiplication modulo p with unit 1), hence a group of order $\text{ord}(G) = p-1$.

Each $x \in G$ generates a subgroup $\langle x \rangle = \{x^1, x^2, \dots\}$ in $G \supset \langle x \rangle$. As G can be represented as disjoint union of the coset classes $g \langle x \rangle$ (with identical cardinality $|g \langle x \rangle|$ for all $g \in G$) we have – as for each subgroup H of a group G –

$$\text{ord}(x) = \text{ord}(\langle x \rangle) \mid \text{ord}(G)$$

For $a \in G$, i.e. relative prime to p we have $v \cdot \text{ord}(a) = \text{ord}(G)$ and thus

$$a^{p-1} = a^{\text{ord}(G)} = a^{v \cdot \text{ord}(a)} = (a^{\text{ord}(\langle a \rangle)})^v = 1^v = 1$$

representing Fermat's little theorem.

$$a^{p-1} \equiv 1 \pmod{p}$$

E.g. $m = 11111$ is not prime because $2^{11110} \equiv 10536 \pmod{11111}$ since

$$\begin{aligned} 2^{15} &\equiv 10546 \pmod{m}, & 2^{90} &\equiv 10546^6 \pmod{m} \equiv 7830 \pmod{m} \\ 2^{150} &\equiv 10546^{10} \pmod{m} \equiv 3771 \pmod{m}, & 2^{310} &\equiv 10536 \pmod{m} \\ 2^{540} &\equiv 7830^6 \pmod{m} \equiv 1 \pmod{m}, & 2^{10800} &= (2^{540})^{20} \equiv 1 \pmod{m} \end{aligned}$$

Equally, $m = 11111$ is not prime because $3^{11110} \equiv 2410 \pmod{11111}$ since

$$\begin{aligned} 3^9 &\equiv 19683 \pmod{m} \equiv 8572 \pmod{m}, & 3^{10} &\equiv 3494 \pmod{m} \\ 3^{60} &\equiv 3494^6 \pmod{m} \equiv 9757 \pmod{m}, & 3^{70} &\equiv 2410 \pmod{m} \\ 3^{120} &\equiv 9757^2 \pmod{m} \equiv 1 \pmod{m}, & 3^{11040} &= (3^{120})^{92} \equiv 1 \pmod{m} \end{aligned}$$

$$\begin{aligned} &\hat{=} & & \pmod{\quad} \\ &\hat{\approx}^{20} & & \end{aligned}$$



²⁰ due to limited accuracy of representation and computation

Problem 18(d)

The implication is equivalent to FLT with $p = n$ and $a = 2$.

$$\text{FLT: } 2^{n-1} \equiv 1 \pmod{n} \Rightarrow 2^{n-1} - 1 \equiv 0 \pmod{n} \Rightarrow n | 2^{n-1} - 1.$$

But, let $n = 341$. Due to $341 = 11 \cdot 31$, n is composite, that is not prime. However $2^{340} \equiv 1 \pmod{341}$, because

$$2^{10} = 3 \cdot 341 + 1 \Rightarrow 2^{10} - 1 = 3 \cdot 341 \Rightarrow 341 | 2^{10} - 1 \Rightarrow 2^{10} \equiv 1 \pmod{341}$$

Taking powers generates a counter example:

$$2^{340} \equiv 1 \pmod{341} \Rightarrow 341 | 2^{340} - 1$$

E.g. 341 is composite because $3^{340} \equiv 56 \pmod{341}$ due to

$$\begin{aligned} 3^6 &= 47 \pmod{341}, 3^7 = 141 \pmod{341}, 3^8 = 82 \pmod{341} \\ 3^9 &= 246 \pmod{341}, 3^{10} = 56 \pmod{341}, 3^{30} = 1 \pmod{341} \\ 3^{330} &= 1 \pmod{341}, \text{ zusammen also } 3^{340} = 56 \pmod{341} \end{aligned}$$



Problem 19(a)

Let $|M|$ denote the cardinality of a set M , i.e. the number of elements of M , then obviously

$$\varphi(p) = |\{m \in \mathbb{N} : m < p, \gcd(m, p) = 1\}| = |\{1, 2, \dots, p-1\}| = p-1$$

$\varphi(n)$ is easily evaluated For small arguments n :

n	$\varphi(n)$	n	$\varphi(n)$	n	$\varphi(n)$
		11	10	21	12
2	1	12	4	22	10
3	2	13	12	23	22
4	2	14	6	24	8
5	4	15	8	25	20
6	2	16	8	26	12
7	6	17	16	27	18
8	4	18	6	28	12
9	6	19	18	29	28
10	4	20	8	30	8



Problem 19(b)

It is to be shown that there are $p^{k-1} - 1$ different $m < p^k$ with a common divisor with p^k , i.e. with at least the divisor p .

$$|\{m = vp : m = vp < p^k\}| = |\{m = vp : 1 \leq v < p^{k-1}\}| = p^{k-1} - 1$$

Hence

$$\begin{aligned}\varphi(p^k) &= |\{1, 2, \dots, p^k - 1\} \setminus \{m = vp : 1 \leq v < p^{k-1}\}| \\ &= |\{1, 2, \dots, p^k - 1\}| - |\{m = vp : 1 \leq v < p^{k-1}\}| \\ &= p^k - 1 - (p^{k-1} - 1) = p^k - p^{k-1}\end{aligned}$$



Problem 19(c) Let $n = r \cdot s$ with relatively prime factors r and s .

The set $M = \{m \in \mathbb{N} : m < n, \gcd(m, n) = 1\}$ can be enumerated as follows: Each r' relatively prime to r and each s' relatively prime to s specifies a $m = r' \cdot s'$ relatively prime to n , i.e.

$$\{r' < r : \gcd(r', r) = 1\} \times \{s' < s : \gcd(s', s) = 1\} \subset M$$

Vice versa, each divisor of $m \in M$ is divisor either of r or of s . Its prime factor decomposition can be thought as a product of two factors relatively prime to either r or to s .

$$\begin{aligned}\varphi(n) &= |M| \\ &= |\{r' < r : \gcd(r', r) = 1\} \times \{s' < s : \gcd(s', s) = 1\}| \\ &= |\{r' < r : \gcd(r', r) = 1\}| \cdot |\{s' < s : \gcd(s', s) = 1\}| \\ &= \varphi(r) \cdot \varphi(s)\end{aligned}$$



Problem 19(d)

Each $n \in \mathbb{N}$ has a prime factor decomposition

$$n = \prod_{i=1}^v p_i^{v_i}$$

where v_i denotes the multiplicity of the prime factor p_i and v the number of prime factors. Therefore

$$\varphi(n) = \prod_{i=1}^v (p_i^{v_i} - p_i^{v_i-1}) = n \prod_{i=1}^v (1 - 1/p_i)$$

but only if the prime factor decomposition of n is known at all.

If p and q are prime and $n = p \cdot q$ then specially

$$\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1) = p \cdot q - p - q + 1 = n - (p+q) + 1$$



Problem 19(e) Proof according to [24] p.57

As in the proof of **Fermat's Little Theorem** let $G = (\mathbb{Z}/n\mathbb{Z})^*$, i.e. the group of the invertible elements of $\mathbb{Z}/n\mathbb{Z}$ with unit $[1] \in \mathbb{Z}/n\mathbb{Z}$. G consists of the elements $[m] \in \mathbb{Z}/n\mathbb{Z}$ whose representants m have a **modulo- n inverse**. These are exactly the elements $[m]$ with representants m relatively prime to n . Hence $G = \{[m] : 1 \leq m < n, \gcd(m, n) = 1\}$ and therefore $\text{ord}(G) = \varphi(n)$.

With $\gcd(a, n) = 1$ is $[a] \in G$ and thus $a^{\text{ord}(G)} = [1]$ or

$$a^{\varphi(n)} = a^{\text{ord}(G)} \equiv 1 \pmod{n}$$



Problem 20(a)

$\gcd(m_i, m_j) = 1$ implies $\gcd(b_i, m_i) = 1$ for $i = 1, \dots, n$.

Therefore, the (modulo m_i)-inverse x_i to b_i exists, i.e.

$$x_i b_i \equiv 1 \pmod{m_i} \quad \text{for } i = 1, \dots, n$$

Also $b_i x_i \equiv 0 \pmod{m_j}$ if $i \neq j$, hence $x_i b_i \equiv \delta_{ij} \pmod{m_i}$. With

$$x = \sum_{i=1}^n x_i b_i r_i$$

we get for $j = 1, \dots, n$

$$x \bmod m_j = \sum_{i=1}^n (x_i b_i r_i) \bmod m_j = (x_j b_j r_j) \bmod m_j = r_j$$



Problem 20(b)

Obviously

$$x = y \pmod{p} \iff x - y \in p\mathbb{Z} \iff p \mid (x - y)$$

$$x = y \pmod{q} \iff x - y \in q\mathbb{Z} \iff q \mid (x - y)$$

Because p and q are by assumption relatively prime, we get

$$(pq) \mid (x - y) \iff x - y \in (pq)\mathbb{Z} \iff x = y \pmod{pq}$$



Problem 20(c) Let a be the age to be computed. Ask for

$$r_1 = a \% 3, \quad a = r_1 \bmod 3, \quad a \equiv r_1 \pmod{3}$$

$$r_2 = a \% 5, \quad a = r_2 \bmod 5, \quad a \equiv r_2 \pmod{5}$$

$$r_3 = a \% 7, \quad a = r_3 \bmod 7, \quad a \equiv r_3 \pmod{7}$$

Then $m = \prod_1^3 m_i = 3 \cdot 5 \cdot 7 = 105$ and additionally $b_1 = 105/3 = 35$, $b_2 = 105/5 = 21$ and $b_3 = 105/7 = 15$. The (modulo m_i)-inverses are

$$x_1 = 2 \text{ because } 2 \cdot 35 = x_1 b_1 = 1 \bmod m_1 = 1 \bmod 3,$$

$$x_2 = 1 \text{ because } 1 \cdot 21 = x_2 b_2 = 1 \bmod m_2 = 1 \bmod 5 \text{ and}$$

$$x_3 = 1 \text{ because } 1 \cdot 15 = x_3 b_3 = 1 \bmod m_3 = 1 \bmod 7.$$

Therefore

$$a = \sum_1^3 x_i b_i r_i \bmod m = (70 r_1 + 21 r_2 + 15 r_3) \bmod 105$$



Problem 21(a) Use $a + 0 = 0 + a = a$ and $a \cdot 1 = 1 \cdot a = a$ to determine all but one result in the addition resp. in the multiplication table.

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} \quad \begin{array}{c|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

If we defined $1 + 1 = 1$, then 1 had no inverse w.r.t. addition.

If we defined $1 \cdot 1 = 0$, then 1 had no inverse w.r.t. multiplication.

We can interpret and implement addition as XOR or as addition modulo 2. We can interpret and implement multiplication as AND or as multiplication modulo 2. □

Problem 21(b) Commutative addition and multiplication in $\mathbb{GF}(3)$

$+$	$ $	0	1	2		\cdot	$ $	0	1	2
0	$ $	0	1	2		0	$ $	0	0	0
1	$ $	1	2	0		1	$ $	0	1	2
2	$ $	2	0	1		2	$ $	0	2	1



Problem 21(c) Commutative addition and multiplication in $\mathbb{GF}(5)$

$+$	0	1	2	3	4	\cdot	0	1	2	3	4
0	0	1	2	3	4	0	0	0	0	0	0
1	1	2	3	4	0	1	0	1	2	3	4
2	2	3	4	0	1	2	0	2	4	1	3
3	3	4	0	1	2	3	0	3	1	4	2
4	4	0	1	2	3	4	0	4	3	2	1



Problem 21(d) For any prime p , addition modulo p makes $\mathbb{GF}(p)$ a commutative group and multiplication modulo p makes $\mathbb{GF}(p)^*$ a commutative group – with distributivity.

$a =$	$a + b =$	$m =$
$b =$	$a \cdot b =$	eval
		reset

However, for $p, q \in \mathbb{GF}(pq)^*$ inadmissably $p \cdot q = 0$ holded, i.e. the product of factors different from zero vanished. □

Problem 22(a) For any $r(x) = \sum_{i=0}^{n-1} r_i x^i \in \mathbb{GF}(p^n)$ and any $s(x) = \sum_{i=0}^{n-1} s_i x^i \in \mathbb{GF}(p^n)$ with $r_i, s_i \in \mathbb{GF}(p)$ define

$$(r + s)(x) := \sum_{i=0}^{n-1} (r_i + s_i) x^i \in \mathbb{GF}(p^n)$$

Then obviously, $(\mathbb{GF}(p^n), +)$ is a commutative (additive) group. Its zero element is the constant polynomial $\text{zero}(x) = 0x^0$. The inverse of a polynomial $q(x) = \sum_{i=0}^{n-1} c_i x^i \in \mathbb{GF}(p^n)$ w.r.t. addition is the polynomial $-q(x) = \sum_{i=0}^{n-1} (p - c_i \bmod p) x^i \in \mathbb{GF}(p^n)$.

$r =$	$p =$
$s =$	$n =$
$r =$	add
$s =$	$s := -r$
$r + s =$	reset



Problem 22(b)

Let $r(x) = \sum_{i=0}^{n-1} r_i x^i \in \mathbb{GF}(p^n)$ and $s(x) = \sum_{i=0}^{n-1} s_i x^i \in \mathbb{GF}(p^n)$ with $r_i, s_i \in \mathbb{GF}(p)$. Then

$$(rs)(x) := \sum_{i=0}^{2n-2} x^i \sum_{j=0}^i r_j s_{i-j} \in \mathbb{P}(2n-1)$$

Because obviously $r \cdot s \in \mathbb{P}(2n-1)$ in general, the product of factors in $\mathbb{GF}(p^n)$ is itself not necessarily in $\mathbb{GF}(p^n)$. \square

Problem 22(c) Then, the polynomial m has to have degree n . A polynomial m of lower degree does not suffice as the following example for $\mathbb{GF}(2^2)$ shows.

$m(x) = x$				
*	0	1	x	$x + 1$
0	0	0	0	0
1	0	1	x	$x + 1$
x	0	x	0	0
$x + 1$	0	$x + 1$	0	1

$m(x) = x + 1$				
*	0	1	x	$x + 1$
0	0	0	0	0
1	0	1	x	$x + 1$
x	0	x	1	0
$x + 1$	0	$x + 1$	0	0

In both cases there are contradictions. In case $m(x) = x$ for example, x has no inverse element. In case $m(x) = x + 1$ for example, $x + 1$ has no inverse element. □

Problem 22(d) Then, m has to be *irreducible*, i.e. m cannot be represented as the product of two non-constant polynomials with lower degree. Namely, assuming $m = m_1 \cdot m_2$ with non-constant m_1 and m_2 of degree not bigger than n . Then $m_1, m_2 \in \mathbb{GF}(p^n)$ and $m_1 \cdot m_2 = 0$ in $\mathbb{GF}(p^n)$ holds. \square

Problem 22(e) m_1 is reducible over $\mathbb{GF}(2)$ because of $m_1(x) = x^2 + 1 = x^2 + 2x + 1 = (x + 1)^2$. Exhaustive listing of all suitable products shows $m_2 = x^2 + x + 1$ to be irreducible over $\mathbb{GF}(2)$.

$$x \cdot x = x^2$$

$$x \cdot (x + 1) = x^2 + x$$

$$(x + 1) \cdot x = x^2 + x$$

$$(x + 1) \cdot (x + 1) = x^2 + 2x + 1 = x^2 + 1$$



Problem 23(a) Multiplication in $\mathbb{GF}(2^2) = \mathbb{GF}(2)[x]/m(x)$ with $m(x) = x^2 + x + 1$ is given by the following table

\cdot	0	1	x	$x + 1$
0	0	0	0	0
1	0	1	x	$x + 1$
x	0	x	$x + 1$	1
$x + 1$	0	$x + 1$	1	x

because

$$x^2 = x + 1 \pmod{m(x)}$$

and

$$x(x + 1) = x^2 + x = x + 1 + x = 1 \pmod{m(x)}$$

and

$$(x + 1)^2 = x^2 + 2x + 1 = x^2 + 1 = x + 2 = x \pmod{m(x)}.$$

The inverse elements can be read directly from the multiplication table. □

Problem 23(b) For $r(x) = \sum_{i=0}^{n-1} r_i x^i \in \mathbb{GF}(p^n)^*$ and $s(x) = \sum_{i=0}^{n-1} s_i x^i \in \mathbb{GF}(p^n)^*$ with $r_i, s_i \in \mathbb{GF}(p)$ define

$$(r \cdot s)(x) := \left(\sum_{i=0}^{n-1} x^i \sum_{j=0}^i r_j s_{i-j} \right) \bmod m(x) \in \mathbb{GF}(p^n)$$

Then, $(\mathbb{GF}(p^n)^*, \cdot)$ is a commutative (multiplicative) group. Its one-element is the constant polynomial $\text{one}(x) = 1x^0$.

$m =$	next irr poly
$r =$	$p =$
$s =$	$n =$
$m =$	irreducible?
$r =$	multiply
$s =$	$s := 1/r$
$r \cdot s =$	reset
$g =$	$ \{\text{irr poly}\} $
	next generator



Problem 23(c) By the extended version of the **Euclid algorithm** on p. 20.

First, the classical algorithm in its recursive and iterative form is presented computing $\gcd(x, y)$.

<pre>gcd_rec(int x,y) { if (y=0) return abs(x); else return gcd_rec(y,mod(x,y)); }</pre>	<pre>gcd_it(int x,y) { int tmp; while (y<>0) { tmp = y; y = mod(x,y); x = tmp; } return abs(x) }</pre>
--	--

The extended Euclid algorithm computes for given x and y coefficients a and b such that $d = \gcd(x, y) = ax + by$.

With prime p and $0 \leq x < p$ then $1 = \gcd(x, p) = ax + bp$ holds, i.e. $ax = 1 - bp$ or $ax = 1 \pmod{p}$. Hence, the extended Euclid algorithm inverts elements in $\mathbb{GF}(p)$ and similar in $\mathbb{GF}(p^n)$.

```
gcd_coeff(int x,y) % returns vector
{
    int q,tmp,q11,q12,q22,t21,t22;
    q11 = q22 = 1;
    q12 = q21 = 0;
    while (y<>0)
    {
        tmp = y;
        q = x / y;
        y = x % y;
        x = tmp;
        t21 = q21; t22 = q22;
        q21 = q11 - q*q21;
        q22 = q12 - q*q22;
        q11 = t21; q12 = t22;
    }
    return vector(x,q11,q12);
}
```



Problem 23(d) Using the feature in the [form](#) on p. [114](#) to generate irreducible polynomials over $\mathbb{GF}(p)$ of a given degree n , the following little table can be established:

$p \setminus n$	2	3	4	5	6	7	8	...
2	1	2	3	6	9	18	30	...
3	6	16	36	96	232	...		
5	40	160	600	...				
⋮	...							

By the way, it can be shown that there always is at least one irreducible polynomial over $\mathbb{GF}(p)$ of degree n .



Problem 23(e) Let $m_1(x)$ and $m_2(x)$ be two irreducible polynomials over $\mathbb{GF}(p)$ of degree $n - 1$ and \mathbb{F}_i the field constructed using $m_i(x)$. These two fields are isomorphic, i.e. except for renaming the elements, the two fields are identical, i.e. there is an isomorphism, a bijective mapping $\varphi : \mathbb{F}_1 \rightarrow \mathbb{F}_2$ with $\varphi(r + s) = \varphi(r) + \varphi(s)$ and $\varphi(r \cdot s) = \varphi(r) \cdot \varphi(s)$

- because any two finite fields with the same number of elements are isomorphic
- or because finite fields are *cyclic*, i.e. for any finite field \mathbb{F} there is a generating element g such that $\mathbb{F}^* = \{g^i; i \in \mathbb{N}\}$. Let g_i be generating element for \mathbb{F}_i . Then $\varphi : \mathbb{F}_1 \rightarrow \mathbb{F}_2$ defined by $\varphi(g_1) := g_2$ and canonically extended to \mathbb{F}_1 , specifies an isomorphism between \mathbb{F}_1 and \mathbb{F}_2 .

□

Problem 23(f) Obviously, $\mathbb{GF}(2^2)^*$ with $m(x) = x^2 + x + 1$ has exactly two generating elements, namely $g_1(x) = x$ and $g_2(x) = x + 1$.

n	$g_1^n = x^n$	n	$g_2^n = (x + 1)^n$
0	$g_1^0 = 1$	0	$g_2^0 = 1$
1	$g_1^1 = x$	1	$g_2^1 = x + 1$
2	$g_1^2 = x + 1$	2	$g_2^2 = x$

$\mathbb{GF}(2^3)^*$ with $m(x) = x^3 + x + 1$ has at least three generating elements: x , $x + 1$ and x^2 .

n	x^n	n	$(x + 1)^n$	n	$(x^2)^n$
0	1	0	1	0	1
1	x	1	$x + 1$	1	x^2
2	x^2	2	$x^2 + 1$	2	$x^2 + x$
3	$x + 1$	3	x^2	3	$x^2 + 1$
4	$x^2 + x$	4	$x^2 + x + 1$	4	x
5	$x^2 + x + 1$	5	x	5	$x + 1$
6	$x^2 + 1$	6	$x^2 + x$	6	$x^2 + x + 1$

$\mathbb{GF}(2^3)^*$ with $m(x) = x^3 + x^2 + 1$ has at least three generating elements:

x , $x + 1$ and x^2 .

n	x^n	n	$(x + 1)^n$	n	$(x^2)^n$
0	1	0	1	0	1
1	x	1	$x + 1$	1	x^2
2	x^2	2	$x^2 + 1$	2	$x^2 + x + 1$
3	$x^2 + 1$	3	x	3	$x^2 + x$
4	$x^2 + x + 1$	4	$x^2 + x$	4	x
5	$x + 1$	5	$x^2 + x + 1$	5	$x^2 + 1$
6	$x^2 + x$	6	x^2	6	$x + 1$

This comes at no surprise: due to isomorphy, the set of generating elements is independent of the choice of the irreducible polynomial m .

There are more generating elements of $\mathbb{GF}(2^3)^*$: namely, the [form](#) on [p. 114](#) computes step by step all generating elements of $\mathbb{GF}(p^n)^*$ for any (small) prime p and (small) $n \in \mathbb{N}$. □

Problem 23(g) Let g be a generating element von $\mathbb{GF}(p^n)^*$ and let $\log r$ be the logarithm of elements $r \in \mathbb{GF}(p^n)^*$ to the base g . Then multiplication can be reduced to addition and three table look ups:

$$(r \cdot s) = g^{\log r + \log s}$$

For example, consider $\mathbb{GF}(2^8)^*$: instead of $256 \cdot 256 = 65536$ entries in a look up table for the multiplication in $\mathbb{GF}(2^8)^*$ only two look up tables with 256 entries each are needed. \square

Problem 23(h) Using

$$1/r = r^{-1} = g^{-\log r} = g^{p^n - 1 - \log r}$$

inversion in $\mathbb{GF}(p^n)^*$ can be implemented by two table look ups and one subtraction. □

Problem 24(a)

With $k = 3$ the message is decrypted to give the plain text

thisisatopsecretmessage.

Obviously there are only 26 possible keys.

Also, the Caesar-encryption/decryption preserves the letter frequencies – a natural angle for an attack.

s.a. www.weblearn.hs-bremen.de/risse/MAI/docs/mai1.pdf

$$k = \quad x =$$

$$\text{Caesar} \quad y =$$

$$\text{Caesar}^{-1} \quad x =$$

? Anything special about this implementation ?



Problem 24(b)

Besides the trivial key 0 there are only 25 other keys $k = 1, 2, \dots, 25$.

Security is rather low since only 25 keys have to be tried. □

Problem 25(a)

Without restricting generality let $k = k \bmod m$.

Encryption has to be one-to-one, i.e.

$$(k x_1) \bmod m = (k x_2) \bmod m \Rightarrow x_1 = x_2 \quad \text{für alle } 0 \leq x_1, x_2 < m$$

or equally

$$x_1 \neq x_2 \Rightarrow (k x_1) \bmod m \neq (k x_2) \bmod m \quad \text{für alle } 0 \leq x_1, x_2 < m$$

Then, necessarily k and m have no common divisor. Assuming otherwise there was some $g \in \mathbb{N}$ with $k = v g$ and $m = w g$. Hence, for $0 = x_1 \neq x_2 = w < m$

$$(k x_1) \bmod m = 0 \bmod m = 0 = (v w g) \bmod (w g)$$

holds in contradiction to encryption being one-to-one. □

Problem 25(b) If and only if $1 = (k k^{\text{inv}}) \bmod m$ then

$$x = (k^{\text{inv}} y) \bmod m = (k^{\text{inv}}((k x) \bmod m)) \bmod m = (k^{\text{inv}} k x) \bmod m$$

for each $0 \leq x < m$. Then k^{inv} is called the *(modulo- m)-inverse* of k .

Now, **Euclid's algorithm** $\text{gcd}(x_o, x_1)$ for some x_o and x_1 with no common divisors computes

$$x_o = q_1 x_1 + x_2 \quad x_2 = x_o - q_1 x_1$$

$$x_1 = q_2 x_2 + x_3 \quad x_3 = x_1 - q_2 x_2 = x_1 - q_2(x_o - q_1 x_1)$$

$$x_2 = q_3 x_3 + x_4 \quad x_4 = x_2 - q_3 x_3 = x_o - q_1 x_1 - q_3(x_1 - q_2(x_o - q_1 x_1))$$

$$\vdots$$

$$x_{n-2} = q_{n-1} x_{n-1} + x_n \quad x_n = \text{linear combination of } x_o \text{ and } x_1$$

$$x_{n-1} = q_n x_n + x_{n+1}$$

until $x_n = 1$ and $x_{n+1} = 0$. Hence, each x_i is a linear combination of x_o and x_1 . Especially for $\text{ggT}(k, m)$ holds $x_n = 1 = u k + v m$ if $\text{gcd}(k, m) = 1$. Therefore follows

$$u k = 1 - v m \Rightarrow (u k) = 1 \bmod m$$

and u is the (modulo m)-inverse k^{inv} of k .

Problem 25(c)

Insertion gives

$$\begin{aligned}x &= (k'_1 y + k'_o) \bmod m = (k'_1((k_1 x + k_o) \bmod m) + k'_o) \bmod m \\ &= (k'_1 k_1 x + k'_1 k_o + k'_o) \bmod m = x\end{aligned}$$

iff and only if

$$k'_1 = k_1^{\text{inv}} \quad \text{and} \quad k'_o = (-k_1^{\text{inv}} k_o) \bmod m$$

Then also

$$\begin{aligned}y &= (k_1 x + k_o) \bmod m = (k_1((k_1^{\text{inv}} y + k'_o) \bmod m) + k_o) \bmod m \\ &= (k_1 k_1^{\text{inv}} y + k_1 k'_o + k_o) \bmod m \\ &= (y + k_1(-k_1^{\text{inv}} k_o) + k_o) \bmod m = y\end{aligned}$$



Problem 25(d)

Each key consists of a pair (k_1, k_o) with $k_o \in \{0, 1, \dots, m - 1\}$ and $k_1 \in \{0 \leq k < m : \gcd(k, m) = 1\}$. Therefore all keys are in the space of keys $\{0, 1, 2, \dots, 25\} \times \{0 \leq k < m : \gcd(k, m) = 1\}$.

For example, in case of $m = 26$ the key space has $26 \cdot \varphi(26) = 26 \cdot 12$ elements.

All the same, the level of security is unchanged and rather low, as the letter frequencies are still preserved.



Problem 26(a)

Using the key word **key** corresponding to $k = 10, 4, 24$, i.e. $l = 3$ the decrypted message is

thisisatopsecretmessage.

Obviously there are 26^l possible keys of length l . This cycle length l can be determined by the method invented by **Kasiski**. Once the cycle length is known an attack consists only of l independent Caesar-decryptations.

$k =$ $x =$

Vigenère $y =$

Vigenère⁻¹ $x =$

? Anything special about this implementation ?



Problem 26(b)

Each key consists of a string k of arbitrary length. Hence, each key is contained in the key space $\cup_{l=1}^{\infty} \{0, 1, 2, \dots, 25\}^l$ if the Latin alphabet is used.

Security of the Vigenère-encryption/decryption scheme is the higher the longer the key. But the longer the key the more difficult it is to transmit a key to all legitimate receivers without the transmission being eavesdropped.

Highest security is achieved – however at the highest cost to transmit the keys – if keys are used only for one time, so called *one time pad*, s.a. www.fourmilab.ch/onetime/otpjs.html □

Problem 27(a)

There are $n! = 1 \cdot 2 \cdot 3 \cdots n = \prod_{i=1}^n i$ permutations of n objects. The Latin alphabet therefore has

$$26! = 403291461126605635584000000$$

permutations.



Problem 27(b)

No, because still the frequency of character combinations is preserved which can be used to decypher an encrypted text.

e.g. Let $y = \underline{C}\underline{E}\underline{S}\underline{V}\underline{L}\underline{R}\underline{H}\underline{E}\underline{E}\underline{S}\underline{U}\underline{U}\underline{S}\underline{L}\underline{L}\underline{G}\underline{A}\underline{N}\underline{O}\underline{S}\underline{G}\underline{M}\underline{I}\underline{H}\underline{R}\underline{S}\underline{T}\underline{U}$ be the encryption of a German text x .

S, C, and H are – because of blocking – close together so that one can guess the trigram SCH. The minimal block length ℓ is 5. For this block length the string x improbably starts with "ELVSCH..." or "EVLSCH...". For $\ell = 6$ SCH is not possible, so this block length is disregarded. For $\ell = 7$ we get

...SCH....SEL....ALG.....HMU...

The final position of the trigrams is not yet known; by using the frequency of other character combinations we get

$x = \text{VERSCHLUESSELUNG\SALGORITHMUS}$

s.a. www.kryptoanalytiker.de



Problem 27(c)

Text blocks of fixed length are encrypted by permutation of its letters. In the following example

htsisitapoesrcteemssga.e

pairs of plain text letters are interchanged, i.e. the permutation $(2, 1)$ is applied to 2-letter blocks.

There are $n!$ permutations of n -letter blocks: the longer the blocks the more permutations or keys there are, i.e. the more secure is the encryption/decryption method. However at the same time the key length grows as well as the cost of buffering messages to be encrypted or decrypted. \square

Problem 28(a)

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

$$\begin{array}{l} x = \\ y = \\ x = \end{array} \qquad \begin{array}{l} \text{padding} \\ \text{DES-}P \\ \text{DES-}P^{\text{inv}} \end{array}$$

? Why is the encoded string represented as a `o|`-string, i.e. each encrypted block of 8 letters as a 64bit block?

Try different padding characters.



Problem 28(b)

Let $f_K^{\text{inv}}(L, R) = (R \oplus K, L)$. Due to $K \oplus K = \vec{0}$ then

$$f_K^{\text{inv}}(f_K(L, R)) = f_K^{\text{inv}}(R, L \oplus K) = (L \oplus K \oplus K, R) = (L, R)$$

as well as

$$f_K(f_K^{\text{inv}}(L, R)) = f_K(R \oplus K, L) = (L \oplus K \oplus K, R) = (L, R)$$

f_K represents a substitution if 64bit blocks (L, R) are considered a letter in the alphabet $\mathcal{A} = \{0, 1\}^{64}$.

$x =$	padding
$K =$	check
$y =$	DES- f
$x =$	DES- f^{inv}

Now, DES consists of 16 such substitutions.

DES encrypts by iterated application of functions f_{K_i} to a message x where the keys K_i are generated from some main key K .

Encryption by

$$\begin{aligned}y &= (P^{\text{inv}} \circ f_{K_{16}} \circ f_{K_{15}} \circ \dots \circ f_{K_2} \circ f_{K_1} \circ P)(x) \\ &= P^{\text{inv}}(f_{K_{16}}(f_{K_{15}}(\dots(f_{K_2}(f_{K_1}(P(x))))\dots)))\end{aligned}$$

implies **Decryption** by

$$\begin{aligned}x &= (P^{\text{inv}} \circ f_{K_1}^{\text{inv}} \circ f_{K_2}^{\text{inv}} \circ \dots \circ f_{K_{15}}^{\text{inv}} \circ f_{K_{16}}^{\text{inv}} \circ P)(y) \\ &= P^{\text{inv}}(f_{K_1}^{\text{inv}}(f_{K_2}^{\text{inv}}(\dots(f_{K_{15}}^{\text{inv}}(f_{K_{16}}^{\text{inv}}(P(y))))\dots)))\end{aligned}$$



Problem 28(c) L can be computed as sequence of matrix transformations and thus is linear. However, linear encryption is relatively easily cracked. \square

Problem 28(d) The look up table for each S-Box has 2^6 lines à 4bit, i.e. $2^8 = 256$ bit, a total of $8 \cdot 256 = 2^{11} = 2$ Kbit for all eight S-boxes. On the other hand, a look up table for a 32bit substitution would have 2^{32} lines à 32bit, i.e. $32 \cdot 4 \cdot 2^{30} = 128$ Gbit – a totally unacceptable alternative.

Use pre-computed inverse S-boxes. □

Problem 29(a) DES keys are 64bit long, including 8 parity bits. Hence, the effective length is 56bit and the key space size is $2^{56} = 64(2^{10})^5 \approx 64(10^3)^5 = 64 \cdot 10^{15}$. \square

Problem 29(b) Only if several DES encryptions cannot be emulated by a single one, i.e. only if

$$\text{DES}_{K_2} \circ \text{DES}_{K_1} \neq \text{DES}_{K_0}$$

holds, then TDEA establishes higher security than DES, s.a.

http://en.wikipedia.org/wiki/Triple_DES



Problem 29(c) TDEA keys are 3×56 bit long.

Hence, the effective length is 168 bit and the key space size is $2^{168} = 256 (2^{10})^{10} \approx 256 (10^3)^{10} = 2.56 \cdot 10^{32}$. □

Problem 29(d) In 2001, the *Advanced Encryption Standard*, *AES* was published and in 2002 standardized. AES is the winner of a public competition.

Correspondingly, the withdrawal of DES resp. TDEA was proposed in 2004 and 2005 finalized. □

Problem 30(a) Public key methods are the more secure the more difficult it is to deduce f_A^{-1} from A and f_A .

Functions f_A with the following properties are suitable for public key encryption/decryption methods:

- f_A is one-to-one. (The plain text is partitioned into fixed length blocks; f_A is applied to each block.)
- f_A and f_A^{-1} are easily evaluated. (Messages are quickly encrypted and decrypted.)
- It is **practically impossible** to deduce f_A^{-1} from f_A . (Encrypted messages can be decrypted only at astronomical cost – at best the decryption cost can be estimated in order to scale the encryption/decryption method according to the security needs.)

By the way, such functions are called *trapdoor functions*. □

Problem 31(a)

The text to be encrypted is partitioned into fixed length blocks so that the (ASCII-) string can be thought of as a (big) $x \in \mathbb{N}$ with $x < n$. Then f_e is applied to each of these x .

Now, f_e is a trapdoor function because

- f_e is one-to-one on $X = \{0, 1, 2, \dots, n - 1\}$, as $f_d = f_e^{-1}$.
Namely

d is modulo- $\varphi(n)$ invers to e , i.e. $de \equiv 1 \pmod{\varphi(n)}$ or $de = v(p-1)(q-1) + 1$ for a $v \in \mathbb{N}$, so that $x^{de} = x \cdot x^{v(p-1)(q-1)}$.

Due to **Fermat's Little Theorem, FLT** for prime p and q

$$\left. \begin{array}{l} x^{p-1} = 1 \pmod{p} \\ x^{q-1} = 1 \pmod{q} \end{array} \right\} \Rightarrow \begin{cases} x^{de} = x(x^{p-1})^{v(q-1)} = x \cdot 1^{v(q-1)} = x \pmod{p} \\ x^{de} = x(x^{q-1})^{v(p-1)} = x \cdot 1^{v(p-1)} = x \pmod{q} \end{cases}$$

From $x^{de} = x \pmod{p}$ and $x^{de} = x \pmod{q}$ follows per **Chinese Remainder Theorem** for p and q with $\gcd(p, q) = 1$, i.e. a fortiori for prime p and q

$$x^{ed} = x \pmod{(pq)} = x \pmod{n}$$

- $f_e(x)$ and $f_d(y)$ resp. are easily evaluated by computing several products $(m_1 \cdot m_2) \bmod n$ for $m_i \in X$.
- The bigger n the more difficult it is to determine f_e , i.e. to infer d from e and n .

For example, in 1994 ca. 600 via Internet networked computers needed a total of **5000MIPS years**, to factorize the 129-digit number R-129²¹ into its two 64- and 65-digit prime factors.

$p =$	$q =$	$n =$	
$e =$	$d =$		check&fill at random
$x =$			
$y =$			RSAenc
$x =$			RSAdec

To check e.g. $\gcd(e, \varphi(n)) = 1$ **Euclids algorithm** is available. □

²¹ D. Atkins, M. Graff, A.K. Lenstra, P.C. Leyland: The magic words are *squeamish ossifrage*; Asiacrypt '94, pp263-277, LNCS 917, Springer 1995

Problem 31(b)

The security of the RSA-method rests on the difficulty to factorize big $n \in \mathbb{N}$ with 100 and more digits.

The RSA-method is the more secure the bigger n , cp. e.g.

[www.comp.mq.edu.au/courses/comp333/Lecture/
factoring_and_RSA_4.pdf](http://www.comp.mq.edu.au/courses/comp333/Lecture/factoring_and_RSA_4.pdf)



Problem 31(c)

Let e_A and e_B be Alice's and Bob's public RSA-key with secret RSA-keys d_A and d_B resp.

Then Alice only has to append to her encrypted message $y = f_{e_B}(x)$ the digital signature $y' = f_{d_A}(x)$.

Bob then decrypts the first half y of the received message to $x = f_{d_B}(y)$ and verifies on the basis of the second half that x and $f_{e_A}(y')$ coincide. As only Alice knows d_A it is only Alice who could have generated y' . Therefore Bob can be assured to have received a message from Alice.

By the way, Alice does not need to use the whole message x to generate the signature $y' = f_{d_A}(x)$. It is sufficient to use a hash-code $\text{hash}(x)$ which both sender and receiver know to generate.

Typical hash-codes are for example **MD4**, **MD5** or **SHA-1**. □

Problem 32(a) AES is a block oriented, symmetrical (identical key for encryption and decryption) encryption/decryption method consisting of rounds of permutations and substitutions.

csrc.nist.gov/publications/fips/fips197/fips-197.pdf



Problem 32(b) AES encrypts 128bit = 16bytes = 4word blocks of plain text. It allows 128bit, 192bit and 256bit keys with 10, 12 or 14 rounds respectively. □

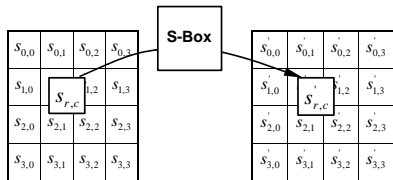
Problem 32(c) An AES encryption round consists of

- substitution of each bytes by another one per s-box
- permutation of the rows of the block when represented as 4×4 -byte-matrix
- permutation of the columns of the block when represented as 4×4 -byte-matrix
- XOR of block and part of the expanded key



Problem 33(a) `SubBytes()`: The substitution of a byte b by the AES s-box is specified to be the **multiplicative inverse** b^{-1} computed in $\mathbb{GF}(2^8)$, followed by the affine transformation

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = b' = Ab + c = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$



The AES s-box is usually implemented as a look up table, i.e.

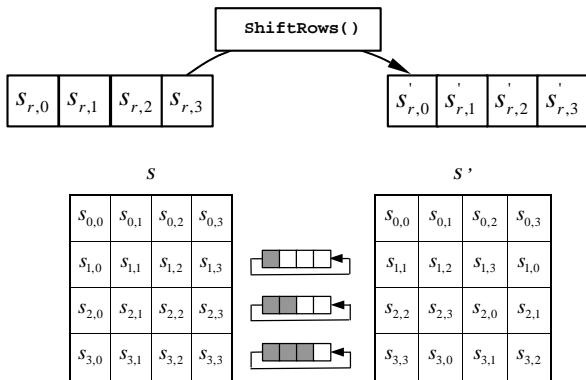
		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	ab	27	b2	75
	4	09	83	2c	1a	1b	6e	5e	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	c5
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

as is the inverse s-box

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d



Problem 33(b) `ShiftRows()`: The rows of a block are cyclically shifted as indicated by the following figure: the block S is thus mapped to the block S' .



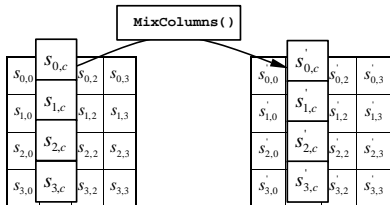
The inverse transformation just shifts rows cyclically in the opposite direction. □

Problem 33(c) `MixColumns()`: The columns of a block are considered as polynomials with coefficients in $\mathbb{GF}(2^8)$ and multiplied by

$$a(x) = 0x03 x^3 + 0x01 x^2 + 0x01 x^1 + 0x02 x^0$$

modulo $x^4 + 1$. The c^{th} column then becomes

$$\begin{pmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{pmatrix} = \begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{pmatrix} \begin{pmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{pmatrix} = A \begin{pmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{pmatrix}$$



The block S is thus mapped to the block S' .

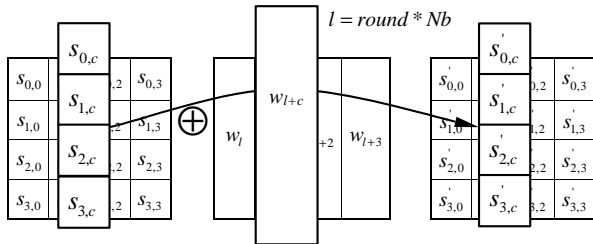
The inverse transformation is specified by multiplication modulo x^4+1 by the polynomial

$$a^{-1} = 0x0b x^3 + 0x0d x^2 + 0x09 x^1 + 0x0e x^0$$

or written as matrix transformation

$$\begin{pmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{pmatrix} = \begin{pmatrix} 0x0e & 0x0b & 0x0d & 0x09 \\ 0x09 & 0x0e & 0x0b & 0x09 \\ 0x0d & 0x09 & 0x0e & 0x0b \\ 0x0b & 0x0d & 0x09 & 0x0e \end{pmatrix} \begin{pmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{pmatrix} = A^{-1} \begin{pmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{pmatrix}.$$

□

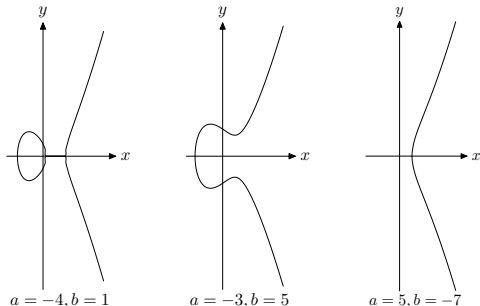
Problem 33(d) AddRoundKey():

As for all XOR operations, this transformation is its own inverse. □

Problem 34(a)

$$E = E(\mathbb{R}) = E_{a,b}(\mathbb{R}) = \left\{ \left(x, \pm \sqrt{x^3 + ax + b} \right) : x^3 + ax + b \geq 0 \right\}$$

Obviously, elliptic curves are plane curves which are symmetric to the x -axis. Depending on the parameters a and b , the radicand is positive in one interval or in two intervals. Correspondingly, $E = E(\mathbb{R}) = E_{a,b}(\mathbb{R})$ has one or two branches. Cp. e.g.



$$\lim_{x \rightarrow +\infty} \pm \sqrt{x^3 + ax + b} = \lim_{x \rightarrow +\infty} \pm x^{3/2} = \pm \infty$$



Problem 34(b)

Because all coefficients of $x^3 + ax + b$ are real, all zeroes x_i can conveniently be represented by trigonometric/hyperbolic means. Let $p = a/3$, $q = b/2$, $D = p^3 + q^2$ and $P = (\text{sgn } q)\sqrt{|p|}$.

	$p < 0, D \leq 0$	$p < 0, D > 0$	$p > 0$
x_1	$\beta = \frac{1}{3} \arccos \frac{q}{P^3}$ $-2P \cos \beta$	$\beta = \frac{1}{3} \operatorname{arcosh} \frac{q}{P^3}$ $-2P \cosh \beta$	$\beta = \frac{1}{3} \operatorname{arsinh} \frac{q}{P^3}$ $-2P \sinh \beta$
$x_{2,3}$	$2P \cos(\beta \pm \pi/3)$	$P(\cosh \beta \pm i\sqrt{3} \sinh \beta)$	$P(\sinh \beta \pm i\sqrt{3} \cosh \beta)$

□

Problem 34(c)

For the given radicand $x^3 + ax + b$ let again $p = a/3$, $q = b/2$ and discriminant $D = p^3 + q^2$. The discriminant D then determines the type of zeroes.

$D > 0$		one real zero, two conjugate complex zeroes
$D < 0$		three distinct real zeroes
$D = 0, q \neq 0$		one simple real, one double real zero
$D = 0, q = 0$		one triple real zero

Hence, there are no multiple zeroes if and only if

$$D = p^3 + q^2 = \frac{a^2}{27} + \frac{b^2}{4} \neq 0$$

or equivalently if

$$108D = 108(p^3 + q^2) = 4a^2 + 27b^2 \neq 0$$



Problem 34(d) Let the line be given by $y = mx + c$ with $m \neq 0$. The abscissa x of an intersection points of the line with E solves $(mx + c)^2 = x^3 + ax + b$ or equivalently

$$x^3 - m^2x^2 + (a - 2cm)x + b - c^2 = 0$$

Substituting $y = x - m^2/3$ the quadratic term is eliminated. According to the assumption the new equation

$$y^3 + 3py + 2q = 0 \quad \text{with} \quad \begin{aligned} 3p &= (a - 2cm) - \frac{1}{3}m^4 \\ 2q &= -\frac{2}{27}m^6 + \frac{1}{3}(a - 2cm)m^2 + b - c^2 \end{aligned}$$

has at least two simple real solutions. According to the **classification** of the solutions in dependence of the discriminant $D = p^3 + q^2$ (on p. 160), to the two simple real solutions there must be another simple real solution. \square

Problem 34(e) Let $s = (y_Q - y_P)/(x_Q - x_P)$ be the slope and hence $y = y(x) = y_P + s(x - x_P)$ the line through P and Q . Then

$$y^2(x) = (y_P + s(x - x_P))^2 = x^3 + ax + b$$

or just

$$x^3 - s^2x^2 + (a + \dots)x + (b + \dots) = 0$$

has the three solutions x_P , x_Q and x_R where x_R is the abscissa of the third intersection point of the line through P and Q with E . Comparison of the coefficients of x^2 gives

$$-s^2 = -x_P - x_Q - x_R \quad \text{or just} \quad x_R = s^2 - x_P - x_Q$$

Mirroring the third intersection point at the x -axis gives $R = P + Q$ so that

$$y_R = -((y_P + s(x_R - x_P))) = s(x_P - x_R) - y_P$$



Problem 35(a) Imagine $P + P$ to be the limit of $P + Q$ with $E \ni Q \rightarrow P$. Then in the limit, the line through P and Q becomes the tangent in P with slope

$$s = \left. \frac{d}{dx} \sqrt{x^3 + ax + b} \right|_{x_P} = \frac{1}{2} \frac{3x_P^2 + a}{\sqrt{x_P^3 + ax_P + b}} = \frac{3x_P^2 + a}{2y_P}$$

Hence, $y = y(x) = y_P + s(x - x_P)$ is the tangent in P . x_P is a double zero of the equation

$$y^2(x) = (y_P + s(x - x_P))^2 = x^3 + ax + b$$

The other simple zero is x_R . Hence, as before, $R = (x_R, y_R) = P + P$ is given by

$$x_R = s^2 - 2x_P \quad \text{and} \quad y_R = -(y_P + s(x_R - x_P)) = s(x_P - x_R) - y_P$$

□

Problem 35(b) Due to the symmetry of E , $Q = -P$ holds. The line through P and Q is vertical and has only these two intersection points with E . Assuming again, that in a limit process $E \ni Q' \rightarrow Q$. Then $R := P + Q'$ moves on the unbounded branch of E towards infinity. Just define this to be an extra point on E , called *the point at infinity* or just 0 .

Using homogeneous coordinates the plane together with the elliptic curve is transformed into projective space which shows that there is only one point at infinity [30]. \square

Problem 35(c) Introduction of 0 as above together with the definition $-P = -(x_P, y_P) := (x_P, -y_P)$ implies that 0 is a neutral or a zero element w.r.t. this addition and that $-P$ is the inverse of P w.r.t. this addition, i.e.

- $P + 0 = 0 + P = P$ for all $P \in E$
in addition, $0 + 0 = 0$
- $P + (-P) = (-P) + P = P - P = 0$ for all $P \in E$
in addition, 0 is inverse to 0

The equation $P + Q = R$ is solved by $Q = (-P) + R$ for any $P, R \in E$.



Problem 35(d)

Because this so defined addition obviously is commutative, it makes $E = E_{a,b}(\mathbb{R})$ an (additive) commutative group or a so called Abel²²ian group. □

²² Niels Henrik Abel (1802-1829)

Problem 36(a) Performing all operations in $\mathbb{GF}(p)$ (cp. [arithmetic in \$\mathbb{GF}\(p\)\$](#) , p. 23) makes $E = E_{a,b}(\mathbb{GF}(p))$ a commutative (additive) group (cp [30] for associativity of this addition).

The neutral element, i.e. the zero element w.r.t. this addition is specified by the point (`<empty string>`,`infty`) here.

Elliptic curve $E(\mathbb{GF}(p)) = \{(x, y) : y^2 = x^3 + ax + b\}$ over $\mathbb{GF}(p)$ with $a =$ $b =$ and $p =$ is a (additive) group with $\text{card}(E(\mathbb{GF}(p))) =$ elements, where by Hasse²³ $|\text{card}(E(\mathbb{GF}(p))) - (p+1)| \leq 2\sqrt{p}$ holds. check

$E(\mathbb{GF}(p)) =$

$P = (x_P, y_P)$ with $x_P =$ <input type="text"/>	$y_P =$ <input type="text"/>	$R := P + Q$
$Q = (x_Q, y_Q)$ with $x_Q =$ <input type="text"/>	$y_Q =$ <input type="text"/>	$Q := -P$
$R = (x_R, y_R)$ with $x_R =$ <input type="text"/>	$y_R =$ <input type="text"/>	<code>c&c</code> ²⁴ reset

□

²³ Helmut Hasse (1898-1979)

www-history.mcs.st-andrews.ac.uk/Biographies/Hasse.html

²⁴ `c&c` = check whether $P, Q \in E$; complete the fields P and Q if necessary

Problem 37(a) In $\mathbb{GF}(2^m)$ any element r is inverse to itself w.r.t. addition, i.e. $-r = r \in \mathbb{GF}(2^m)$. Hence $P = (x, y)$ with $y^2 = x^3 + ax + b$ and $-P = (x, -y)$ were identical in $E = E_{a,b}(\mathbb{GF}(2^m))$, and $2P = P + P = P - P = 0$ for any $P \in E$, so that E is isomorphic to $\mathbb{GF}(2) \times \mathbb{GF}(2) \times \dots \times \mathbb{GF}(2)$.

Therefore, the subgroups generated by any element of E have only two elements preventing any usage in cryptographic applications (cp. *discrete logarithm-problem*). \square

Problem 37(b) Performing all operations in $\mathbb{GF}(2^m)$ (cp. [arithmetic in \$\mathbb{GF}\(pn\)\$](#) , p. 25) makes $E = E_{a,b}(\mathbb{GF}(2^m))$ a commutative (additive) group. \square

Problem 38(a) ECC is a block oriented, asymmetrical public key encryption/decryption method using the group structure on elliptic curves $E = E_{a,b}(\mathbb{F})$ over $\mathbb{F} = \mathbb{GF}(p)$ or $\mathbb{F} = \mathbb{GF}(2^m)$.

There is an EC encryption/decryption (*ECIES*), an EC Diffie-Hellman key exchange (*ECDH*), and an EC digital signature algorithm (*ECDSA*).

Due to its superior performance ECC is mainly used to replace RSA in hybride encryption/decryption schemes.

[29]

[31] www.secg.org/collateral/sec1_final.pdf

[30] www.iaik.tugraz.at/.../oswald/papers/Introduction_to_ECC.pdf

S.a. e.g. www.faqs.org/rfcs/rfc3278.html,

<http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-5/No-1&2/ROBERTS.pdf>



Problem 38(b) Communication partners agree on some elliptic curve $E = E(\mathbb{F})$ over some finite field \mathbb{F} together with some suitable generator point $G \in E$. Let $n = \text{card}(\langle G \rangle)$. Each partner chooses some random number $0 < r < n$ as secret key and publishes rG as public key.

	chooses	publishes
Alice	a	$Q_A = aG$
Bob	b	$Q_B = bG$
\vdots	\vdots	\vdots

In order to encrypt and send a message m to Bob, Alice converts the message to a point $M \in E$, chooses some random number k and sends the encrypted message, i.e. the pair $(kG, M + k(bG)) \in E \times E$ to Bob.

	chooses	encrypts	decrypts
Alice	k	$(kG, M + kQ_B)$	
Bob		$(kG, M + k(bG))$	$M = M + kbG - b(kG)$

To decrypt $(kG, M + k(bG))$, Bob computes $M + k(bG) - b(kG) = M$.



Problem 38(c) Before exchanging a common secret key, Alice and Bob agree on a public elliptic curve $E = E(\mathbb{F})$ over some finite field \mathbb{F} together with some generator point $G \in E$.

Let $n = \text{card}(\langle G \rangle)$.

Now, each partner chooses some random number $r \in \mathbb{N}$ with $1 < r < n$ as secret key, publishes the corresponding public key $Q = rG \in E$ and computes a secret key $R \in E$.

	chooses	publishes	computes
Alice	a	$Q_A = aG$	$R_A = aQ_B$
Bob	b	$Q_B = bG$	$R_B = bQ_A$

Because of

$$R_A = aQ_B = abG = baG = bQ_A = R_B$$

Alice and Bob share the same secret $R_A = R = R_B$, the common secret key R . □

Problem 38(d) Let $n = \text{card}(\langle G \rangle)$. Alice wants to sign message m to Bob. Her secret key is $a \in \mathbb{N}$ and her public key is $Q = aG \in E$.

	chooses	hashes	computes	signs
Alice	k	$e = \text{hash}(m)$	$r = x_{kG} \bmod n$ $h = k^{-1} \bmod n$ $s = h(e + ar) \bmod n$	(r, s)

Alice repeats choosing some $1 < k < n$ until $r \neq 0$ and $s \neq 0$.

Bob receives Alice's message m together with her signature (r, s) .

	hashes	computes	verifies
Bob	$e = \text{hash}(m)$	$w = s^{-1} \bmod n$ $u = ew \bmod n, v = rw \bmod n$ $P = uG + vQ$	$x_P == r$

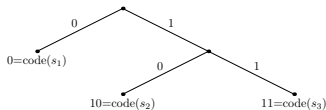
$s = k^{-1}(e + ar) \bmod n \iff k = s^{-1}(e + ar) \bmod n$. Thus, modulo n

$$k \equiv s^{-1}(e + ar) \equiv s^{-1}e + s^{-1}ar \equiv we + war \equiv u + arw \equiv u + va$$

so that $P = uG + vQ = uG + vaG = (u + va)G = kG$ and hence $x_P = x_{kG} = r$ follows. \square

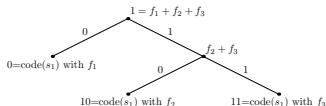
Problem 39(a) First, this coding takes symbol frequencies into account: the more frequent a symbol the shorter its code. Second, because of $\text{code}(s_i) \subset \text{code}(s_j)$ für $i < j$, this coding is not prefix-free. Third, the symbol 0 acts as a separator of codes. Presumably, there must be better codings. □

Problem 39(b) A coding can be represented by a labelled graph with a root: the set $\{\text{code}(s_i) : i = 1, \dots, n\}$ of codes is just the set of labels of its end vertices, i.e. vertices with exactly one incident edge.



Obviously, a coding is prefix-free if and only if the graph representing this coding is a binary tree. □

Problem 39(c) The codes of a prefix-free coding are the leaves of its representing tree. Label the leaf vertices by the corresponding frequencies. Each internal vertex is root of exactly one subtree. Its label is just the sum of the labels of all other vertices of its subtree.



Now, if only the symbol frequencies are given, the tree has to be built starting from the leaves. In the example above, for $c_1 = \text{code}(s_1)$ to be shortest, necessarily $f_1 \geq f_2 + f_3$ holds. This can be generalized:

David A. Huffman: A method for the construction of minimum-redundancy codes; Proceedings of the Institute of Radio Engineers, I.R.E. Sept 1952, S. 1098-1102 <http://compression.ru/download/articles/huff/huffman.1952.minimum-redundancy-codes.pdf>



Problem 40(a)

To simplify matters, the alphabet consists of say 64 characters blank (ASCII 32) up to underline (ASCII 95).

```

text
TEXT
pat=      chr=                init LZWstep
  TXT
|dict| =   dict[      ]=      check  reset


---


codes
old=      new=                init WZLstep
TEXT
|dict| =   dict[      ]=      check  reset

```

? Anything special about this implementation ?



Problem 40(b) The modified decompression of the algorithm:

```
Read OLD_CODE
```

```
CHARACTER = dict[oldCODE]; output CHARACTER
```

```
WHILE there are still input characters DO
```

```
    Read newCODE
```

```
    IF newCODE is not in dictionary
```

```
        PATTERN = dict[oldCODE]
```

```
        PATTERN = PATTERN+CHARACTER
```

```
    ELSE
```

```
        PATTERN = dict[newCODE]
```

```
    END of IF
```

```
    output PATTERN
```

```
    CHARACTER = first character in PATTERN
```

```
    add dict[oldCODE] + CHARACTER to dictionary
```

```
    oldCODE = newCODE
```



Problem 41(a) There are three possible cases, namely RR, RR und RB.

There are two favorable cases, namely RB.

Therefore, $P = P(RB) = 1/3$.



Problem 41(b) Let a/A and z/Z indicate a door with a car resp. a goat behind. Small letters correspond to initially chosen doors.

Without loss of generality assume that the candidate chooses door no 1, and the quizmaster reveals the goat behind door no 2. Then

chances to win without revision: $P(aZZ) = 1/3$

chances to win with revision: $P(zZA \text{ or } zZA) = 2/3$

www.comedia.com/hot/monty.html or (*Monte-Carlo-*) experiment:

doors	left	middle	right	
state (A=car, Z=goat)				1 ×
x=choice, o=revelation				10 ×
				100 ×
				reset

A total of hits ^{without} _{with} revisions in a total of games, i.e.

chances to win \approx ^{without} _{with} revisions □

Problem 42(a) Discriminating features are

- data type and co-domain, e.g.
 - 0-1-sequences, e.g. coin tosses,
 - natural or integer random numbers, e.g. decimal digits of π ,
 - rational random numbers, e.g. measured distances of darts-arrows to the middle of the disk,
 - real random numbers, e.g. freie Weglänge of particles in Brownian motion, etc.
- distribution of the random numbers in their co-domain, e.g.
 - evenly distributed 0-1-random numbers, e.g. tossing a true coin,
 - Poisson-distributed natural random numbers, e.g. number of radioactive decays per time unit,
 - exponentially distributed random numbers, e.g. life time of non-aging parts,
 - normal distributed random numbers with mean μ and standard deviation σ , e.g. physical measurements, etc.

The continuous random Variable $X \in [0, 1]$, evenly distributed in the unit interval, is a suitable standard-random variable: from X one generates by

if ($X \leq 0.5$) **return 0; else return 1;**

evenly distributed discrete random numbers $Y \in \{0, 1\}$,

if ($X < p_1$) **return** y_1 ;

if ($X < p_1 + p_2$) **return** y_2 ;

\vdots

if ($X < p_1 + \dots + p_n$) **return** y_n ;

discrete random numbers $Y \in \{y_1, y_2, \dots, y_n\}$ with $P(Y = y_i) = p_i$ für $i = 1, 2, \dots, n$,

(b-a)*X+a

in the interval $[a, b] \subset \mathbb{R}$ evenly distributed, continuous random numbers Y ,

round((b-a)*X+a)

in the interval $[a, b] \cap \mathbb{Z}$ evenly distributed, discrete random numbers Y , etc.

For in the unit interval evenly distributed continuous random numbers $X \in \mathbb{R}$, $F^{\text{inv}}(X)$ generates continuous random numbers $Y = F^{\text{inv}}(X)$ with a given distribution function F and its inverse function F^{inv} because $P(Y < y) = P(F^{\text{inv}}(X) < y) = P(X < F(y)) = F(y)$.



Problem 42(b) There are a number of algorithms to generate pseudo random numbers. All procedures are recursive, well known is e.g. J. v. Neumanns *method of middle digits of squares*

$$x_{n+1} = (x_n^2)_{3b\dots b-1}$$

for suitable $2b$ bit x_o where $(x_n^2)_{3b\dots b-1}$ denotes the middle $2b$ bit of the $4b$ bit product x_n^2 – or better and more commonly used

$$x_{n+1} = a x_n \pmod{m}$$

for some x_o , say $x_o = 1$, for a suitable factor a of magnitude 2^b and for a modulus $m = 2^b$ if b is the integer width of the computer and if efficiency is at premium. This generator is a special case of the so called *Linear Congruential Generators*

$$x_{n+1} = (a x_n + c) \pmod{m}$$

for some $x_o \in \mathbb{N}$, say $x_o = 1$, and suitable parameters $a, c, m \in \mathbb{N}$. □

Problem 42(c) $x_{n+1} \in \{0, 1, \dots, m-1\}$. Hence, the maximal periodic length is m . For $a = 1$ and $c = 0$ it is 1. \square

Problem 43(a) Histogramming shows to what degree random numbers cover the given co-domain. This is tested by the following simulation: As here in JavaScript $b = 64$, choose m of magnitude 2^{32} , a of magnitude 2^{16} and some $0 \leq c < m$. Then random numbers

$$y_n = 2^{-r} x_n \quad \text{where} \quad x_{n+1} = (a x_n + c) \pmod{m}$$

are generated and the relative frequency of their occurrence in certain intervalls is monitored. Let $h_i = \mathbf{round}(100 P(Y = 2^{-r} X \in [\frac{i-1}{5}, \frac{i}{5}]))$.

$a =$		$c =$		$m =$	
$h_1 =$	%	$h_2 =$	%	$h_3 =$	%
$h_4 =$	%	$h_5 =$	%		%
$n =$		$1 \times$	$10 \times$	$100 \times$	test
					reset

□

Problem 43(b) The entropy

$$E = - \sum_{i=0}^9 p_i \log_2(p_i)$$

i.e. the information content of each decimal digit (bit per decimal digit, bpdd), is maximal for true random numbers (with independent digits). The entropy E is (for evenly distributed digits) maximal

$$E_{\max} = - \sum_{i=0}^9 \frac{1}{10} \log_2\left(\frac{1}{10}\right) = - \log_2\left(\frac{1}{10}\right) \approx 3.321928 \text{ bpdd}$$

$a =$

$c =$

$m =$

$n =$

$x =$

$E =$

$1 \times$

$10 \times$

$100 \times$

test

reset



Problem 43(c) A sequence of pseudo random numbers can be compressed whereby the lower compression rate the higher the degree of unpredictability. Let the compressability κ with $0 \leq \kappa \leq 1$ be defined by

$$\kappa = \frac{\text{length of compressed pseudo random number sequence}}{\text{length of uncompressed pseudo random number sequence}}$$

using for example Huffman coding. Compressability κ is maximal 1 for true random numbers. \square

Problem 43(d) The statistical χ^2 -test checks whether two random variables are statistically independent. (It is distribution free, i.e. the distributions of the two variables do not matter.)

It could be applied to check the independence of pairs (x, y) of members of a sequence $(x_i)_{i=0,1,\dots}$ of random numbers, say (x_i, x_{i+1}) or somewhat more general (x_i, x_{i+d}) for fixed $d \in \mathbb{N}$. Here, let x be the **decimal digits** of the pseudo random numbers and $d = 1$.

Therefore we need to set up the so called *contingency table*.

$x \backslash y$	0	1	...	j	...	9	$f_{i,*}$	absolute frequency
0	$f_{0,0}$	$f_{0,1}$...	$f_{0,j}$...	$f_{0,9}$	$f_{0,*}$	$f_{i,j} = \{(i, j)\} $
1	$f_{1,0}$	$f_{1,1}$...	$f_{1,j}$...	$f_{1,9}$	$f_{1,*}$	and absolute mar-
\vdots	\vdots	\vdots		\vdots		\vdots		ginal frequencies
i	$f_{i,0}$	$f_{i,1}$...	$f_{i,j}$...	$f_{i,9}$	$f_{i,*}$	with $f_{i,*} = \sum_{j=0}^9 f_{i,j}$
\vdots	\vdots	\vdots		\vdots		\vdots		and
9	$f_{9,0}$	$f_{9,1}$...	$f_{9,j}$...	$f_{9,9}$	$f_{9,*}$	$f_{*,j} = \sum_{i=0}^9 f_{i,j}$
$f_{*,j}$	$f_{*,0}$	$f_{*,1}$...	$f_{*,j}$...	$f_{*,9}$	$n = \#$	of observations

$a =$ $x \backslash y$	0	1	2	$c =$ 3	4	5	6	$m =$ 7	8	9	$f_{i,*}$
0											
1											
2											
3											
4											
5											
6											
7											
8											
9											
$f_{*,j}$											

$\alpha =$ $x =$ $\chi^2 =$
 $1 \times$ $10 \times$ $100 \times$ test reset

Of course, the test can be applied to the sequence of pseudo random numbers for any $d \in \mathbb{N}$. Additionally, the sequence can be considered as a bit string. in order to apply the test to pairs of bit substrings of any given length in any given distance. \square