Cryptography

Universität Wuppertal, SS 2018
Universität Braunschweig, WS 2007
Universität Siegen, WS 2005
Universität Duisburg-Essen, SS 2004

Jürgen Müller

Contents

I Classical cryptography

1 Basic cryptographical notions

(1.1) Cryptographic goals. Cryptography is the study of mathematical techniques related to information security. Basic objectives are as follows:

a) Confidentiality, **privacy** or **secrecy**: Keeping information private to those who are authorised to see it. • **Ownership**: A means to provide an entity with the legal right to use or transfer a resource to others. • **Anonymity**: Concealing the identity of an identity involved to some process. • **Access control**: Restricting access to resources to privileged entities. • **Authorisation**: Conveyance to another entity of official sanction to do or be something. • **Receipt**: Acknowledgement that information has been received. • **Confirmation**: Acknowledgement that services have been provided.

b) Data integrity: Ensuring information has not been altered by unauthorised or unknown means. • **Certification**: Endorsement of information by a trusted authority. • **Validation**: A means to provide timeliness of authorisation to use or manipulate information or resources. • **Time stamping**: Recording the time of creation or existence of information.

c) Authentication: • **Identification** or **entity authentication**: Corroboration of the identity of an entity; for example, a person, a computer terminal, a credit card. • **Message authentication** or **data origin authentication**: Corroborating the source of information. • **Signature**: A means to bind information to an entity. • **Witnessing**: Verifying the creation or existence of information by an entity other than the creator. • **Non-repudiation**: Preventing the denial of previous commitments or actions. • **Revocation**: Retraction of certification or authorisation.

d) Practical aspects: • **Level of security**: The number of operations needed to defeat an information security objective. • **Functionality**: Effective combinations of tools to meet various information security objectives. • **Performance**: Efficiency of a cryptographical tools; for exmaple, the number of bits an encryption algorithm can encrypt per time unit. • **Ease of implementation**: Software and hardware complexity.

(1.2) Cryptosystems. a) A **cryptosystem** or **cipher** $[\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D}]$ is a tuple,a where the **plaintexts** $\mathcal{P}$, the **ciphertexts** $\mathcal{C}$ and the **keys** $\mathcal{K}$ are finite sets, and where $\mathcal{E} = \{E_e \colon \mathcal{P} \to \mathcal{C}; e \in \mathcal{K}\}$ and $\mathcal{D} = \{D_d \colon \mathcal{C} \to \mathcal{P}; d \in \mathcal{K}\}$ are **encryption** and **decryption functions**, respectively, such that for all $e \in \mathcal{K}$ there is $d \in \mathcal{K}$ such that $E_e D_d = \mathrm{id}_{\mathcal{P}}$.

The idea is to keep information private to communication partners, Alice and Bob say, who communicate through an insecure channel, where data might be caught by an opponent, Oscar say. Hence plaintexts are first encrypted by Bob, then sent through the channel, and are decrypted again by Alice. Thus in practice, given the keys, encryption and decryption functions should be efficiently computable.

It should be difficult for Oscar to determine plaintexts from ciphertexts without knowing the keys used. Since by **Kerckhoff's principle** the cryptosystem used is assumed to be public, it should also be difficult for Oscar to determine the keys employed.

For a given encryption key $e \in \mathcal{K}$ there might be several suitable decryption keys $d \in \mathcal{K}$. Anyway, from $E_e D_d = \mathrm{id}_{\mathcal{P}}$ we conclude that E_e is injective, which makes sense since otherwise unique decryption would be impossible. Similarly, any (useful) decryption function D_d is surjective. Hence in the case $\mathcal{P} = \mathcal{C}$ encryption functions and (useful) decryption functions are permutations.

b) If, given an encryption key $e \in \mathcal{K}$, a suitable decryption key $d \in \mathcal{K}$ can be assumed to be equal to e, or if d can be easily computed from e, then the cryptosystem is called **symmetric** or a **private-key cryptosystem**. In this case Alice and Bob first have to exchange the keys securely, by applying a **key management** technique.

If a suitable decryption key d cannot be computed easily from e, then the cryptosystem is called **asymmetric** and can be used as a **public-key cryptosystem**: To receive messages Alice publishes $e \in \mathcal{K}$, which Bob uses to encrypt messages, but Alice keeps the suitable decryption keys $d \in \mathcal{K}$ private. In this case no secure key exchange is necessary.

(1.3) Cryptanalysis. The possibility to determine plaintexts from ciphertexts, without finding the keys employed, is called **protocol failure**. The possibility to determine the keys employed is called to **breaking** the cryptosystem. Basic attacks are as follows; according to the varying capabilities Oscar might have the former two are called **passive** and the latter two are called **active** attacks:

a) In a **ciphertext-only attack** Oscar only knows a ciphertexts. For example, it might be based on a statistical analysis of the plaintext language.

For example, we might use **exhaustive search**: Oscar decrypts the ciphertext with all possible keys, and finds the correct plaintext amongst those few which make sense. How large a set of keys might be to be susceptible to exhaustive search, depends on the computing power available.

For example, the DES is based on $2^{56} \sim 7 \cdot 10^{16}$ keys, which nowadays is completely searchable in a couple of days of CPU time. The AES allows for various sets of keys, of size $2^{128} \sim 3 \cdot 10^{38}$, or $2^{192} \sim 3 \cdot 10^{57}$, or $2^{256} \sim 10^{77}$. As a rule of thumb, in practice sets of keys should not have less than 2^{128} elements.

b) In a **known-plaintext attack** Oscar knows plaintext-ciphertext pairs. For example, many letters end with `sincerely yours`, and Oscar might easily intercept the corresponding ciphertext.

c) In a **chosen-plaintext attack** Oscar is able to encrypt plaintexts of his choice, enabling him to obtain plaintext-ciphertext pairs. For example, this is always possible in a public-key cryptosystem.

d) In a **chosen-ciphertext attack** Oscar has the possibility to decrypt ciphertexts of his choice, without knowing the decryption key. Depending on whether Oscar chooses ciphertexts according to ciphertexts observed or not, the attack is called **adaptive** or **indifferent**.

For example, in public-key cryptosystems used in identification protocols the following is possible: Bob wants to convince himself that he is communicating with Alice, by sending her an encrypted random plaintext as a challenge, which Alice decrypts using her private key, and then returns the plaintext to Bob. The opponent Oscar might impersonate Bob and send ciphertexts of his choice to Alice.

(1.4) Security. There are various measures of the security of a cryptosystem, from the most stringent to the weakest security level, and of course depending on the kind of attack launched:

a) Perfect or **unconditional security.** Assuming Oscar has unlimited computational resources, it is impossible to collect any information about plaintexts; the notion of **information** is made precise in information theory, using ideas from stochastics. **Semantic security** refers to the possibility to collect information about plaintexts using algorithms running in expected polynomial time.

b) Computational or **complexity-theoretical security.** Given a model of computing, Oscar is allowed to launch attacks using polynomial time algorithms, in terms of the input size of the cryptosystem, and asymptotic worst case analysis is carried out. Hence care is needed if the theoretical analysis is intended to have practical significance, for example if the average case or at least certain cases are substantially easier than the worst case, or if computations possible in the model in practice are unfeasible.

c) Provable security. In a reduction process it is shown that algorithmically forcing a protocol failure or breaking the cryptosystem is at least as difficult as algorithmically solving another problem. Hence the security of the cryptosystem depends on the actual difficulty of the reference problem.

d) Practical security. The best known algorithms leading to a protocol failure or breaking the cryptosystem need an amount of computing time which by a sufficient margin exceeds the computational resources Oscar might have. This depends on the state of the art of algorithms and machinery.

For example, the number of words of length 64 over the binary alphabet is $2^{64} = 18\,446\,744\,073\,709\,551\,616 \sim 1.8 \cdot 10^{19}$, while a year has $365 \cdot 24 \cdot 3600 = 31\,536\,000 \sim 3 \cdot 10^7$ seconds, hence a CPU having a clock frequency of 3 GHz $= 3 \cdot 10^9$ Hz performs $94\,608\,000\,000\,000\,000 \sim 9.5 \cdot 10^{16}$ clock cycles per year.

e) Heuristic security. This encompasses any variety of arguments that forcing a protocol failure or breaking the cryptosystem needs a prohibitively large amount of practical computational resources; for example, the longer the cryptosystem is in use and is investigated the higher its heuristic security becomes.

(1.5) Signatures. Being used to authenticate the sender of a message, signatures behave differently from conventional signatures, in the following respects:

a) Since no conventional hard copy of the message is signed, the signature has to be explicitly bound to the message. This is achieved by using the message in the verification process. Alternatively, we the message can also be incorporated into the signature, from which the former can then be recovered.

b) Conventional signatures are verified by authorised identities only, by comparing the signature with a known private authentic one. Since the verification algorithm is considered to be public, everyone is able to verify a signature, hence it has to be made sure that forgeries are easy to find. Another possibility is to require the sender of a message to participate in the verification process.

c) Copies of a conventional message can be distinguished from the original message. Since message-signature pairs are identical to their copies, it has to be made sure that a valid pair can only be used once. This can also be achieved by requiring the sender of a message to participate in the verification process.

d) Since a sender could later on deny his signature, a possibility is needed to prove that a signature actually is a forgery, possibly under participation of the sender; then the disavowing sender could just be asked to prove that the signature is non-valid.

e) If a signed message serves as a legal document, the signature scheme must also be secure in the future. In this respect security precautions for signature schemes have to higher than those for cryptosystems.

2 Basic mathematical notions

(2.1) Symmetric groups. a) Let $\mathcal{S}_X := \{\pi \colon X \to X; \pi \text{ bijective}\}$, where X is a set, together with composition of maps be the **symmetric group** on X; the elements of $\mathcal{S}_X$ are called **permutations**. In particular, for $X = \{1, \ldots, n\}$ where $n \in \mathbb{N}_0$ we let $\mathcal{S}_n := \mathcal{S}_{\{1,\ldots,n\}}$; hence for $n = 0$ we have $\mathcal{S}_0 = \{\mathrm{id}_\emptyset\}$.

For $n \in \mathbb{N}_0$ we have $|\mathcal{S}_n| = n!$, as is seen by induction: For $n = 0$ we have $|\mathcal{S}_0| = 1$. For $n \geq 1$ and $\pi \in \mathcal{S}_n$ we have $n\pi = m$ for some $m \in \{1, \ldots, n\}$, hence $\pi \colon \{1, \ldots, n-1\} \to \{1, \ldots, n\} \setminus \{m\}$ is a bijection. There are n possibilities to choose m, hence there are $n \cdot |\mathcal{S}_{n-1}| = n \cdot (n-1)! = n!$ possibilities for π.

E. g. for $n = 1$ we have $\mathrm{id}_{\{1\}}$, for $n = 2$ we have $\mathrm{id}_{\{1,2\}}$ and $\pi \colon 1 \mapsto 2, 2 \mapsto 1$, and for $n = 3$ we have the following permutations, where in the second line we record the images of the elements in the first line:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{bmatrix}.$$

Due to bijectivity, any permutation in $\mathcal{S}_n$ can be written as a product of **disjoint cycles**, where the cycles occurring are uniquely determined up to reordering and rotating. Hence to describe permutations we use **cycle notation**, where 1-cycles are left out: E. g. we have $\mathcal{S}_1 = \{()\}$, and $\mathcal{S}_2 = \{(), (1,2)\}$, and $\mathcal{S}_3 = \{(), (1,2), (1,3), (2,3), (1,2,3), (1,3,2)\}$.

While $\mathcal{S}_1$ and $\mathcal{S}_2$ are commutative, we from $(1,2,3) \cdot (1,2) = (2,3) \neq (1,3) = (1,2) \cdot (1,2,3)$ deduce that $\mathcal{S}_n$ is not commutative for $n \geq 3$. Inverses are given by reading cycles backwardly: E. g. we have $(1,2,3)^{-1} = (1,3,2)$ and $(1,3,2)^{-1} = (1,2,3)$, while the other elements of $\mathcal{S}_3$ are their own inverses.

b) For $k \geq 2$ we have $(a_1, a_2, \ldots, a_k) = (a_1, a_2)(a_1, a_3) \cdots (a_1, a_k)$, which is a product of $k-1$ **transpositions**, i. e. 2-cycles. Hence any cycle and thus any permutation is a product of transpositions. This representation is not necessarily unique, not even the number of transpositions is; e. g. we have $(1,2,3) = (1,2)(1,3) = (1,3)(2,3) = (1,2)(2,3)(1,3)(1,2) \in \mathcal{S}_3$. For $n \in \mathbb{N}$ we only have the following: If $\pi = \tau_1 \cdots \tau_s \in \mathcal{S}_n$ is a product of $r \in \mathbb{N}$ disjoint cycles, where the $\tau_i \in \mathcal{S}_n$ are transpositions and $s \in \mathbb{N}_0$, then $s \equiv n-r \pmod 2$:

We proceed by induction on $s \in \mathbb{N}_0$: For $s = 0$ we have $\pi = ()$ and $r = n$. For $s > 0$ let $\tau_s = (i,j) \in \mathcal{S}_n$ and $\sigma := \tau_1 \cdots \tau_{s-1} \in \mathcal{S}_n$. Let σ be a product of $r' \in \mathbb{N}$ disjoint cycles, by induction we have $s - 1 \equiv n - r' \pmod 2$. If i, j occur in the same cycle of σ, then $\pi = \sigma\tau_s = (\ldots)(i, \ldots, k, j, l, \ldots)(i,j) = (\ldots)(i, \ldots, k)(j, l, \ldots)$. Hence π is a product of $r = r' + 1$ disjoint cycles, and $n - r \equiv n - r' - 1 \equiv s - 2 \equiv s \pmod 2$. If i, j occur in distinct cycles of σ, then we have $\pi = \sigma\tau_s = (\ldots)(i, \ldots, k)(j, \ldots, l)(i,j) = (\ldots)(i \ldots k, j, \ldots, l)$. Hence π is a product of $r = r' - 1$ disjoint cycles, and $n - r \equiv n - r' + 1 \equiv s \pmod 2$. $\sharp$

Thus the **sign** map $\text{sgn}\colon \mathcal{S}_n \to \{\pm 1\}\colon \pi = \tau_1 \cdots \tau_s \mapsto (-1)^s$ is a group homomorphism. The elements in $\text{sgn}^{-1}(1) \subseteq \mathcal{S}_n$ and $\text{sgn}^{-1}(-1) \subseteq \mathcal{S}_n$ are called **even** and **odd** permutations, respectively.

(2.2) Residue class rings. a) Let $n \in \mathbb{N}$ be a **modulus**. For $x \in \mathbb{Z}$ let $\overline{x} := \{y \in \mathbb{Z}; n \mid x - y\} = \{y \in \mathbb{Z}; y \equiv x \pmod n\} \subseteq \mathbb{Z}$ be the associated **residue class**, and let $\mathbb{Z}/n\mathbb{Z} := \{\overline{x} \subseteq \mathbb{Z}; x \in \mathbb{Z}\}$. This induces an equivalence relation on $\mathbb{Z}$, hence $\mathbb{Z}$ is the disjoint union of the distinct residue classes. Quotient and remainder shows that each residue class contains precisely one element from $\mathbb{Z}_n := \{0, \ldots, n-1\}$, hence $\mathbb{Z}_n$ is a set of **representatives** of the residue classes, given by the bijection $\mathbb{Z}_n \to \mathbb{Z}/n\mathbb{Z}\colon x \mapsto \overline{x}$. Thus all the following considerations can be transported from $\mathbb{Z}/n\mathbb{Z}$ to $\mathbb{Z}_n$.

Addition and multiplication on $\mathbb{Z}$ induce operations on $\mathbb{Z}/n\mathbb{Z}$ by $\overline{x} + \overline{y} := \overline{x+y}$ and $\overline{x} \cdot \overline{y} := \overline{xy}$. Using the properties of the addition on $\mathbb{Z}$ we conclude that the addition on $\mathbb{Z}/n\mathbb{Z}$ is associative, commutative, has the neutral element $\overline{0}$, and has additive inverses $-\overline{x} = \overline{-x}$, for $x \in \mathbb{Z}$; thus $\mathbb{Z}/n\mathbb{Z}$ is a commutative additive group. Similarly, the multiplication on $\mathbb{Z}/n\mathbb{Z}$ is associative, commutative, and has the neutral element $\overline{1}$; thus $\mathbb{Z}/n\mathbb{Z}$ is a commutative multiplicative monoid. Finally, distributivity holds as well; thus $\mathbb{Z}/n\mathbb{Z}$ is a commutative ring, being called the associated **residue class ring**.

b) An element $\overline{x} \in \mathbb{Z}/n\mathbb{Z}$ is called a **unit**, if there is $\overline{y} \in \mathbb{Z}/n\mathbb{Z}$ such that $\overline{xy} = \overline{1} \in \mathbb{Z}/n\mathbb{Z}$. The set of units $(\mathbb{Z}/n\mathbb{Z})^*$ is a commutative multiplicative group, being called the **group of units** or the **group of prime residues**. In particular the inverse $\overline{x}^{-1} = \overline{y} \in (\mathbb{Z}/n\mathbb{Z})^*$ of $\overline{x} \in (\mathbb{Z}/n\mathbb{Z})^*$ is unique, and $\mathbb{Z}/n\mathbb{Z}$ is a field if and only if $(\mathbb{Z}/n\mathbb{Z})^* = (\mathbb{Z}/n\mathbb{Z}) \setminus \{\overline{0}\}$.

For $\overline{x} \in \mathbb{Z}/n\mathbb{Z}$ we have $\overline{x} \in (\mathbb{Z}/n\mathbb{Z})^*$ if and only if $\gcd(x, n) = 1$: If $\overline{x} \in (\mathbb{Z}/n\mathbb{Z})^*$, then $\overline{xy} = \overline{1}$ implies that there are $s, t \in \mathbb{Z}$ such that $sxy + tn = 1$, hence $\gcd(x, n) = 1$; conversely, if $\gcd(x, n) = 1$ then there are **Bézout coefficients** $s, t \in \mathbb{Z}$ such that $sx + tn = 1$, which implies $\overline{sx} = \overline{1}$. Thus inverses can be computed by the **Euclidean algorithm**; and $\mathbb{Z}/n\mathbb{Z}$ is a field if and only if n is a prime, in this case we also write $\mathbb{F}_n := \mathbb{Z}/n\mathbb{Z}$.

c) The function $\varphi\colon \mathbb{N} \to \mathbb{N}\colon n \mapsto |(\mathbb{Z}/n\mathbb{Z})^*|$ is called **Euler's totient function**. Letting $p \in \mathbb{N}$ be a prime and $a \in \mathbb{N}$, then we have $|(\mathbb{Z}/p^a\mathbb{Z}) \setminus (\mathbb{Z}/p^a\mathbb{Z})^*| = |\{x \in \{0, \ldots, p^a - 1\}; \gcd(x, p^a) \neq 1\}| = |\{x \in \{0, \ldots, p^a - 1\}; p \mid x\}| = |\{px; x \in \{0, \ldots, p^{a-1}\}| = p^{a-1}$, and thus $\varphi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1)$. This allows us to evaluate φ in general, since φ is a **number theoretic function**, i. e. whenever $m, n \in \mathbb{N}$ are such that $\mathrm{ggT}(m, n) = 1$, then $\varphi(mn) = \varphi(m)\varphi(n)$:

The set $(\mathbb{Z}/m\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})$ can be equipped with componentwise addition and multiplication, and thus becomes a commutative ring. We have the **Chinese remainder theorem** saying that $\alpha\colon \mathbb{Z}/mn\mathbb{Z} \to (\mathbb{Z}/m\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})\colon x + mn\mathbb{Z} \mapsto [x + m\mathbb{Z}, x + n\mathbb{Z}]$ is a ring isomorphism: The map α is a well-defined ring homomorphism, having kernel $\ker(\alpha) = \{x + mn\mathbb{Z}; x + m\mathbb{Z} = 0 + m\mathbb{Z}, x + n\mathbb{Z} = 0 + n\mathbb{Z}\} = \{x + mn\mathbb{Z}; m \mid x, n \mid x\} = \{x + mn\mathbb{Z}; mn \mid x\} = \{0 + mn\mathbb{Z}\}$, thus is injective, and since $|\mathbb{Z}_{mn}| = mn = |\mathbb{Z}_m \times \mathbb{Z}_n|$ it is bijective. Thus we have a group isomorphism $\alpha\colon (\mathbb{Z}/mn\mathbb{Z})^* \to (\mathbb{Z}/m\mathbb{Z})^* \times (\mathbb{Z}/n\mathbb{Z})^*$. ♯

The Chinese remainder theorem is constructive: The Euclidean algorithm yields $s, t \in \mathbb{Z}$ such that $sm + tn = 1$. For given $y + m\mathbb{Z} \in \mathbb{Z}/m\mathbb{Z}$ and $z + n\mathbb{Z} \in \mathbb{Z}/n\mathbb{Z}$ we let $x + mn\mathbb{Z} := (1 - sm)y + (1 - tn)z + mn\mathbb{Z} \in \mathbb{Z}/mn\mathbb{Z}$, and get $\alpha(x + mn\mathbb{Z}) = [(1 - sm)y + (1 - tn)z + m\mathbb{Z}, (1 - sm)y + (1 - tn)z + n\mathbb{Z}] = [(1 - sm)y + smz + m\mathbb{Z}, tny + (1 - tn)z + n\mathbb{Z}] = [y + m\mathbb{Z}, z + n\mathbb{Z}]$.

(2.3) Integral domains. a) Let $R \neq \{0\}$ be a commutative ring. An element $0 \neq a \in R$ is called a **zero-divisor** if there is $0 \neq b \in R$ such that $ab = 0$. If R does not contain zero-divisors it is called an **integral domain**. E. g. since a unit is not a zero-divisor, any field K is an integral domain; $\mathbb{Z}$ is an integral domain; the polynomial ring $K[X]$ is an integral domain.

b) Let R be an integral domain. Then $a \in R$ is called a **divisor** of $b \in R$, and b is called a **multiple** of a, if there is $c \in R$ such that $ac = b$; we write $a \mid b$. Elements $a, b \in R$ are called **associate**, if there is a unit $u \in R^*$ such that $a = bu \in R$; we write $a \sim b$.

Let $\emptyset \neq S \subseteq R$ be a subset. Then $b \in R$ such that $b \mid a$ for all $a \in S$ is called a

common divisor of S. If moreover for any common divisor $c \in R$ of S we have $c \mid b$, then $b \in R$ is called a **greatest common divisor** of S. Let $\gcd(S) \subseteq R$ be the set of all greatest common divisors of S; greatest common divisors are pairwise associate, but do not necessarily exist. Elements $a, b \in R$ such that $\gcd(a, b) = R^*$ are called **coprime**.

Let $0 \neq a \in R \setminus R^*$. Then a is called **irreducible** or **indecomposable**, if $a = bc$ implies $b \in R^*$ or $c \in R^*$ for all $b, c \in R$. Moreover, a is called a **prime**, if $a \mid bc$ implies $a \mid b$ or $a \mid c$ for all $b, c \in R$. If a is a prime, then it is irreducible: Let $a = bc$ for some $b, c \in R$, where we may assume $a \mid b$, then $b = ad$ for some $d \in R$, hence $a = acd$ implies $a(1 - cd) = 0$, and thus $cd = 1$, i. e. $c \in R^*$.

R is called **factorial** or **Gaussian**, if any $0 \neq a \in R$ can be written uniquely, up to reordering and taking associates, in the form $a = u \cdot \prod_{i=1}^r p_i \in R$, where the $p_i \in R$ are irreducible, $r \in \mathbb{N}_0$ and $u \in R^*$. Collecting with respect to associate classes we write $a = u \cdot \prod_{i \in \mathcal{I}} p_i^{\nu_i(f)}$, where the $p_i \in R$ are pairwise non-associate and irreducible, $\nu_i(a) \in \mathbb{N}_0$ are the associated **multiplicities**, $\mathcal{I}$ is a finite index set and $u \in R^*$. If $\nu_i(a) \leq 1$ for all $i \in \mathcal{I}$, then a is called **squarefree**. Given $0 \neq a, b \in R$ then $\gcd(a, b) = R^* \cdot \prod_{i \in \mathcal{I}} p_i^{\min\{\nu_i(a), \nu_i(b)\}}$.

If R is factorial, then an irreducible element $a \in R$ is a prime: Let $0 \neq b, c \in R$ such that $a \mid bc$. Hence there is $a' \in R$ such that $aa' = bc = u \cdot \prod_{i \in \mathcal{I}} p_i^{\nu_i(b)+\nu_i(c)}$. Since a is irreducible, uniqueness of factorisation implies $a \sim p_i$ for some $i \in \mathcal{I}$ such that $\nu_i(b) + \nu_i(c) > 0$, hence $a \mid b$ or $a \mid c$.

c) An integral domain R is called **Euclidean**, if it has a **degree function** $\delta \colon R \setminus \{0\} \to \mathbb{N}_0$ having the following property: For all $a, b \in R$ such that $b \neq 0$ there are $q, r \in R$, called **quotient** and **remainder** respectively, such that $a = qb + r$ where $r = 0$ or $\delta(r) < \delta(b)$.

E. g. a field K is Euclidean with respect to the degree function $K \setminus \{0\} \to \mathbb{N}_0 \colon x \mapsto 0$; and $\mathbb{Z}$ with respect to $\mathbb{Z} \setminus \{0\} \to \mathbb{N}_0 \colon z \mapsto |z|$; and $K[X]$ with respect to $K[X] \setminus \{0\} \to \mathbb{N}_0 \colon f \mapsto \deg(f)$. All these additionally fulfil $\delta(a) \leq \delta(ab)$ for all $0 \neq a, b \in R$, and $\delta(a) < \delta(ab)$ if moreover $b \notin R^*$.

If R is Euclidean, then it is factorial, hence for any $a, b \in R$ there is $r \in \gcd(a, b) \subseteq R$. Moreover, r can be computed by the **Euclidean algorithm**, together with **Bézout coefficients** $s, t \in R$ such that $r = sa + tb$.

(2.4) Alphabets. a) A finite set $\mathcal{X} \neq \emptyset$ is called an **alphabet**, its elements are called **letters** or **symbols**, and $|\mathcal{X}| \in \mathbb{N}$ is called its **length**. A finite sequence w consisting of $n \in \mathbb{N}$ letters is called a **word** over $\mathcal{X}$ of **length** $l(w) = n$. The empty sequence ϵ is called the **empty word**, and we let $l(\epsilon) := 0$.

Let $\mathcal{X}^n$ be the set of all words of length $n \in \mathbb{N}_0$, let $\mathcal{X}^{\leq m} := \coprod_{n \in \{0,\ldots,m\}} \mathcal{X}^n$ and $\mathcal{X}^{\geq m} := \coprod_{n \in \mathbb{N}, n \geq m} \mathcal{X}^n$ for $m \in \mathbb{N}_0$, and let $\mathcal{X}^* := \coprod_{n \in \mathbb{N}_0} \mathcal{X}^n$. For $v, w \in \mathcal{X}^*$ let $vw \in \mathcal{X}^*$ be their **concatenation**. We have $l(vw) = l(v) + l(w)$ and $v\epsilon = \epsilon v = v$ and $(uv)w = u(vw)$, for $u, v, w \in \mathcal{X}^*$. Hence $\mathcal{X}^*$ is a monoid, the **free monoid** over $\mathcal{X}$. A subset $\mathcal{L} \subseteq \mathcal{X}^*$ is called a **(formal) language**.

b) To encrypt and decrypt plain English texts, say, we use the alphabet $\mathcal{X}_{\text{latin}} = \{\texttt{a}, \ldots, \texttt{z}\}$ of Latin letters. These letters are **encoded** into and **decoded** from the alphabet $\mathbb{Z}_{26} := \{0, \ldots, 25\}$ as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

In computer science, the binary alphabet $\mathbb{Z}_2 := \{0, 1\}$ of length 2 is used. For interchange of written texts the ASCII alphabet of length 128, being encoded into and decoded from $\mathbb{Z}_{128} := \{0, \ldots, 127\}$, is used, see [2, Tbl.3.2].

3 Substitution ciphers

(3.1) Substitution ciphers. A cryptosystem such that $\mathcal{P} = \mathcal{C} = \mathcal{X}$, where $\mathcal{X}$ is an alphabet, is called a **substitution** or **monoalphabetic cipher** over $\mathcal{X}$. Most generally we let $\mathcal{K} := \mathcal{S}_{\mathcal{X}}$ be the symmetric group on $\mathcal{X}$, and let the encryption and decryption functions associated with $\pi \in \mathcal{S}_{\mathcal{X}}$ be defined as $E_\pi \colon \mathcal{X} \to \mathcal{X} \colon v \to v\pi$ and $D_\pi \colon \mathcal{X} \to \mathcal{X} \colon v \to v\pi^{-1}$. Thus we have $E_\pi D_\pi = \mathrm{id}_{\mathcal{X}}$, hence substitution ciphers are symmetric cryptosystems.

We have $|\mathcal{K}| = |\mathcal{S}_{\mathcal{X}}| = |\mathcal{X}|!$, hence exhaustive search attacks are not feasible: For example, for $|\mathcal{X}| = 26$ we get $|\mathcal{S}_{\mathcal{X}}| = 26! \sim 4 \cdot 10^{26} \sim 2^{88}$. But it is inefficient to store and evaluate arbitrary permutations in $\mathcal{S}_{\mathcal{X}}$. Thus we might restrict to suitable subsets of permutations in $\mathcal{S}_{\mathcal{X}}$, an extreme case being the following:

(3.2) Shift ciphers. A **shift cipher** is given as follows: Let $\mathcal{P} = \mathcal{C} = \mathcal{K} := \mathbb{Z}_n$, where $n \in \mathbb{N}$, and for $k \in \mathbb{Z}_n$ let $E_k \colon \mathbb{Z}_n \to \mathbb{Z}_n \colon x \mapsto x + k$ and $D_k \colon \mathbb{Z}_n \to \mathbb{Z}_n \colon x \mapsto x - k$. We have $E_k D_k = \mathrm{id}_{\mathbb{Z}_n}$; for example, the **Caesar cipher** is reported to be the case $k = 3$. A shift cipher can be interpreted as a substitution cipher, where $\mathcal{K} = \{(0, 1, \ldots, n-1)^k \in \mathcal{S}_{\mathbb{Z}_n}; k \in \mathbb{Z}_n\}$.

For example, let $n = 26$, and encoding $\mathcal{X}_{\text{latin}}$ into $\mathbb{Z}_{26}$, the word `cryptography` is encoded letter by letter into the plaintext $[2, 17, 24, 15, 19, 14, 6, 17, 0, 15, 7, 24]$, using $k = 11$ yields the ciphertext $[13, 2, 9, 0, 4, 25, 17, 2, 11, 0, 18, 9]$, which is decoded into `ncjaezrclasj`. To find the key k used, we launch a ciphertext-only attack by exhaustive search, running through all possible keys, and indeed recover $k = 11$ as the only sensible solution, see Table 1.

(3.3) Breaking substitution ciphers. We describe a ciphertext-only attack using statistical properties of the plaintext language, here English:

Table 2 gives the probability p_i, for $i \in \mathbb{Z}_{26}$, of occurrence of the various letters in $\mathcal{X}_{\text{latin}}$, Thus `e` occurs most often, with probability 0.127, and [`t`,`a`,`o`,`i`,`n`,`s`,`h`,`r`]

Table 1: Breaking a shift cipher.

k	$D_k([13,2,9,0,4,25,17,2,11,0,18,9])$												
0	13	2	9	0	4	25	17	2	11	0	18	9	`ncjaezrclasj`
1	12	1	8	25	3	24	16	1	10	25	17	8	`mbizdyqbkzri`
2	11	0	7	24	2	23	15	0	9	24	16	7	`lahycxpajyqh`
3	10	25	6	23	1	22	14	25	8	23	15	6	`kzgxbwozixpg`
4	9	24	5	22	0	21	13	24	7	22	14	5	`jyfwavnyhwof`
5	8	23	4	21	25	20	12	23	6	21	13	4	`ixevzumxgvne`
6	7	22	3	20	24	19	11	22	5	20	12	3	`hwduytlwfumd`
7	6	21	2	19	23	18	10	21	4	19	11	2	`gvctxskvetlc`
8	5	20	1	18	22	17	9	20	3	18	10	1	`fubswrjudskb`
9	4	19	0	17	21	16	8	19	2	17	9	0	`etarvqitcrja`
10	3	18	25	16	20	15	7	18	1	16	8	25	`dszquphsbqiz`
11	2	17	24	15	19	14	6	17	0	15	7	24	`cryptography`
12	1	16	23	14	18	13	5	16	25	14	6	23	`bqxosnfqzogx`
13	0	15	22	13	17	12	4	15	24	13	5	22	`apwnrmepynfw`
14	25	14	21	12	16	11	3	14	23	12	4	21	`zovmqldoxmev`
15	24	13	20	11	15	10	2	13	22	11	3	20	`ynulpkcnwldu`
16	23	12	19	10	14	9	1	12	21	10	2	19	`xmtkojbmvkct`
17	22	11	18	9	13	8	0	11	20	9	1	18	`wlsjnialujbs`
18	21	10	17	8	12	7	25	10	19	8	0	17	`vkrimhzktiar`
19	20	9	16	7	11	6	24	9	18	7	25	16	`ujqhlgyjshzq`
20	19	8	15	6	10	5	23	8	17	6	24	15	`tipgkfxirgyp`
21	18	7	14	5	9	4	22	7	16	5	23	14	`shofjewhqfxo`
22	17	6	13	4	8	3	21	6	15	4	22	13	`rgneidvgpewn`
23	16	5	12	3	7	2	20	5	14	3	21	12	`qfmdhcufodvm`
24	15	4	11	2	6	1	19	4	13	2	20	11	`pelcgbtencul`
25	14	3	10	1	5	0	18	3	12	1	19	10	`odkbfasdmbtk`

have probability between 0.091 and 0.060 in decreasing order, while the other letters have probability at most 0.043. The most frequent pairs and triples ordered with respect to decreasing probability are [`th,he,in,er,an,re,ed,on,es,st,en,at,to,nt,ha,nd,ou,ea,ng,as,or,ti,is,et,it,ar,te,se,hi,of`] as well as [`the,ing,and`], respectively.

E. g. we consider the following ciphertext of length 168, and try to find the key $\pi \in \mathcal{S}_{\mathcal{X}_{\text{latin}}}$ used and to determine the plaintext:

```
yifqfmzrwqfyvecfmdzpcvmrzwnmdzvejbtxcddumj
ndifefmdzcdmqzkceyfcjmyrncwjcszrexchzunmxz
nzucdrjxyysmrtmeyifzwdyvzvyfzumrzcrwnzdzjj
xzwgchsmrnmdhncmfqchzjmxjzwiejyucfwdjnzdir
```

The frequency of occurrence of the various letters is given in Table 3. Since `z` occurs much more often than any other ciphertext letter, we conjecture that

Table 2: Probability of letters.

$\mathcal{X}_{\text{latin}}$	$\mathbb{Z}_{26}$	p_i
a	0	0.082
b	1	0.015
c	2	0.028
d	3	0.043
e	4	0.127
f	5	0.022
g	6	0.020
h	7	0.061
i	8	0.070
j	9	0.002
k	10	0.008
l	11	0.040
m	12	0.024

$\mathcal{X}_{\text{latin}}$	$\mathbb{Z}_{26}$	p_i
n	13	0.067
o	14	0.075
p	15	0.019
q	16	0.001
r	17	0.060
s	18	0.063
t	19	0.091
u	20	0.028
v	21	0.010
w	22	0.023
x	23	0.001
y	24	0.020
z	25	0.001

$D_\pi(\texttt{z}) = \texttt{e}$. Since the eight ciphertext letters occurring most often, that is at least nine times each, in decreasing order are {m, c, d, f, j, r, y, n}, we conjecture that the latter decrypt to {t, a, o, i, n, s, h, r}. Since zw occurs four times, and w is rare, we conjecture that $D_\pi(\texttt{w}) = \texttt{d}$. Since rw occurs twice, and r is frequent, we conjecture that $D_\pi(\texttt{r}) = \texttt{n}$. Since nz occurs three times, while zn occurs only once, and n is frequent, we conjecture that $D_\pi(\texttt{n}) = \texttt{h}$. Hence we have:

$\mathcal{C}$	z	w	r	n
$\mathcal{P}$	e	d	n	h

```
......end.........e....nedh..e............
h.......e....e.........nh.d...en....e.h..e
he...n......n......ed...e...e..ne.ndhe.e..
.ed.....nh...h......e....ed.......d..he..n
```

Since c is frequent, the plaintext word ne.ndhe leads to conjecture $D_\pi(\texttt{c}) = \texttt{a}$. Since m is frequent, using mr, which occurs four times, we conjecture that $D_\pi(\texttt{m}) \in \{\texttt{a, i, o}\}$ is a vowel, and since cm occurs once, where ai is much more likely than ao, we conjecture that $D_\pi(\texttt{m}) = \texttt{i}$. Hence we have:

$\mathcal{C}$	z	w	r	n	c	m
$\mathcal{P}$	e	d	n	h	a	i

```
.....iend.....a.i.e.a.inedhi.e......a...i.
h.....i.ea.i.e.a...a.i.nhad.a.en..a.e.hi.e
he.a.n.....in.i....ed...e...e.ineandhe.e..
.ed.a..inhi..hai..a.e.i..ed.....a.d..he..n
```

Table 3: Frequency of letters.

$\mathcal{X}_{\text{latin}}$	frequency	$\mathcal{X}_{\text{latin}}$	frequency
a	0	n	9
b	1	o	0
c	15	p	1
d	13	q	4
e	7	r	10
f	11	s	3
g	1	t	2
h	4	u	5
i	5	v	5
j	11	w	8
k	1	x	6
l	0	y	10
m	16	z	20

From the frequent letters, we have $E_\pi(\texttt{o}) \in \{\texttt{d, f, j, y}\}$ left. If $E_\pi(\texttt{o}) \neq \texttt{y}$, then one of the ciphertexts `cdm, cfm, cjm`, which do occur, leads to a triple of vowels, which is unlikely. Hence we conjecture $D_\pi(\texttt{y}) = \texttt{o}$. Thus from the frequent letters we have $D_\pi(\texttt{d}) \in \{\texttt{t, s, r}\}$ left. Since `nmd` occurs twice, we conjecture that $D_\pi(\texttt{d}) = \texttt{s}$. Since `f` is frequent, and the the plaintext word `.hai.` could be completed to `chair`, this leads to the conjecture $D_\pi(\texttt{f}) = \texttt{r}$ and $D_\pi(\texttt{h}) = \texttt{c}$. Since `j` is the last remaining frequent letter, this implies $D_\pi(\texttt{j}) = \texttt{t}$. Hence we have:

$\mathcal{C}$	z	w	r	n	c	m	y	d	f	h	j
$\mathcal{P}$	e	d	n	h	a	i	o	s	r	c	t

```
o.r.riend.ro..arise.a.inedhise..t...ass.it
hs.r.riseasi.e.a.orationhadta.en..ace.hi.e
he.asnt.oo.in.i.o.redso.e.ore.ineandhesett
.ed.ac.inhischair.aceti.ted..to.ardsthes.n
```

Now a fraction of $\frac{127}{168} \sim 0.76$ of all ciphertext letters has been decrypted. It is thus easy to complete the plaintext, and to determine the key $\pi \in \mathcal{S}_{\mathcal{X}_{\text{latin}}}$, up to those letters which do not occur at all in the plaintext:

```
our friend from paris examined his empty glass wit
h surprise as if evaporation had taken place while
he wasnt looking i poured some more wine and he sett
led back in his chair face tilted up towards the sun
```

4 Block ciphers

(4.1) Block ciphers. A cryptosystem such that $\mathcal{P} = \mathcal{C} = \mathcal{X}^l$, where $\mathcal{X}$ is an alphabet and $l \in \mathbb{N}$, is called a **block** or **polyalphabetic cipher** over $\mathcal{X}$ of **block length** l; a block cipher of block length 1 is a substitution cipher. Most generally we let $\mathcal{K} := \mathcal{S}_{\mathcal{X}^l}$ be the symmetric group on $\mathcal{X}^l$, and let the encryption and decryption functions associated with $\pi \in \mathcal{S}_{\mathcal{X}^l}$ be defined as $E_\pi \colon \mathcal{X}^l \to \mathcal{X}^l \colon v \to v\pi$ and $D_\pi \colon \mathcal{X}^l \to \mathcal{X}^l \colon v \to v\pi^{-1}$. Thus we have $E_\pi D_\pi = \mathrm{id}_{\mathcal{X}^l}$, hence block ciphers are symmetric cryptosystems.

We have $|\mathcal{K}| = |\mathcal{S}_{\mathcal{X}^l}| = (|\mathcal{X}|^l)!$, hence exhaustive search attacks are not feasible. But it is inefficient to store and evaluate arbitrary permutations in $\mathcal{S}_{\mathcal{X}^l}$. Thus we restrict to suitable subsets of permutations in $\mathcal{S}_{\mathcal{X}^l}$:

(4.2) Permutation ciphers. A **permutation cipher** is a block cipher over $\mathcal{X}$ of block length l, where $\mathcal{K} := \mathcal{S}_l$ and where for $\pi \in \mathcal{S}_l$ we let $E_\pi \colon \mathcal{X}^l \to \mathcal{X}^l \colon [v_1, \ldots, v_l] \mapsto [v_{1\pi^{-1}}, \ldots, v_{l\pi^{-1}}]$ as well as $D_\pi \colon \mathcal{X}^l \to \mathcal{X}^l \colon [v_1, \ldots, v_l] \mapsto [v_{1\pi}, \ldots, v_{l\pi}]$. Thus encryption and decryption only permutes the positions of the letters in a block, and we have $|\mathcal{K}| = |\mathcal{S}_n| = l!$.

E. g. let $\mathcal{X} = \mathbb{Z}_{26}$ and $l = 6$, as well as $\pi := (1,3)(2,6,4,5) \in \mathcal{S}_6$, and thus $\pi^{-1} = (1,3)(2,5,4,6) \in \mathcal{S}_6$. Hence the word `she sells sea shells by the sea shore` is cut into blocks of length 6, yielding `shesel lsseas hellsb ythese ashore`. This is encoded block by block into plaintext in $\mathbb{Z}_{26}^6$ and ciphertext in $\mathbb{Z}_{26}^6$ is obtained as follows:

	v	$E_\pi(v)$	
`shesel`	$[18, 7, 4, 18, 4, 11]$	$[4, 4, 18, 11, 18, 7]$	`eeslsh`
`lsseas`	$[11, 18, 18, 4, 0, 18]$	$[18, 0, 11, 18, 4, 18]$	`salses`
`hellsb`	$[7, 4, 11, 11, 18, 1]$	$[11, 18, 7, 1, 11, 4]$	`lshble`
`ythese`	$[24, 19, 7, 4, 18, 4]$	$[7, 18, 24, 4, 4, 19]$	`hsyeet`
`ashore`	$[0, 18, 7, 14, 17, 4]$	$[7, 17, 0, 4, 14, 18]$	`hraeos`

(4.3) Operation modes of block ciphers. To explain the operation modes of block ciphers used in practice, we consider a block cipher over the alphabet $\mathcal{X} := \mathbb{Z}_2 = \{0, 1\}$ of block length l. Recall that there are $(2^l)!$ keys, of which $l!$ are **bit permutations**, amongst which l are **circular right bit shifts**.

a) The **electronic codebook mode (ECB)** is the most basic one: Plaintexts are just cut into blocks of length l, and encrypted and decrypted block by block; if necessary random letters are added to obtain a word of length divisible by l. Since equal plaintext blocks have equal encryptions, the allows for ciphertext-only attacks using statistical properties of the plaintext language.

For example, for block length $l := 4$, and using the key $\pi := (1, 2, 3, 4) \in \mathcal{S}_4$ in a permutation cipher, that is a circular right bit shift, the plaintext word `0001 0100 1010` yields the ciphertext word `1000 0010 0101`.

b) In **cipherblock chaining mode (CBC)** encryption of a block also depends on previous ciphertext blocks. Thus this avoids the weakness of the ECB mode mentioned above, and actually allows for using it for message authentication.

In order to proceed, we define an addition $+ \colon \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ and (for later use) a multiplication $\cdot \colon \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ on $\mathbb{Z}_2$ as follows; note that by identifying 0 and 1 with false and true, respectively, these amount to the logical exclusive or (xor) and and operations, respectively:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Now let $\pi \in \mathcal{S}_{\mathbb{Z}_2^l}$ be a key, and let $w_0 \in \mathbb{Z}_2^l$ be an **initialisation vector**, which can be made public. Given a sequence of plaintext blocks $[v_1, v_2, \ldots] \subseteq \mathbb{Z}_2^l$, we recursively obtain the associated sequence of ciphertext blocks $[w_1, w_2, \ldots] \subseteq \mathbb{Z}_2^l$ as $w_i := E_\pi(v_i \oplus w_{i-1})$, for $i \in \mathbb{N}$, where addition $\oplus$ of vectors is performed entrywise. To decrypt, let $v_i := D_\pi(w_i) \oplus w_{i-1}$, for $i \in \mathbb{N}$, where we indeed have $D_\pi(E_\pi(v_i \oplus w_{i-1})) \oplus w_{i-1} = v_i \oplus w_{i-1} \oplus w_{i-1} = v_i$.

Hence both encryption and decryption need xor operations, and an application of a permutation, where the former computationally are fast operations, while the latter is expensive. Since decryption has to take place completely after encryption, this protocol might be too slow in real-time applications in which messages have to be decrypted while receiving them.

For example, still letting $l := 4$, using the key $\pi := (1, 2, 3, 4) \in \mathcal{S}_4$ in a permutation cipher, and using the initialisation vector $w_0 := [1, 0, 1, 0]$, the plaintext word 0001 0100 1010 yields the ciphertext word 1101 1100 0011, where encryption and decryption are given as follows:

i	v_i	$v_i \oplus w_{i-1}$	w_i
0			$[1, 0, 1, 0]$
1	$[0, 0, 0, 1]$	$[1, 0, 1, 1]$	$[1, 1, 0, 1]$
2	$[0, 1, 0, 0]$	$[1, 0, 0, 1]$	$[1, 1, 0, 0]$
3	$[1, 0, 1, 0]$	$[0, 1, 1, 0]$	$[0, 0, 1, 1]$

c) To circumvent the above-mentioned weaknesses, **keystreams** are used as an auxiliary tool. To describe how this is done, we need a few preparations first:

We may view the vector $v = [x_{k-1}, \ldots, x_0] \in \mathbb{Z}_2^k$ as the **binary representation** of the integer $\widehat{v} := \sum_{i=0}^{k-1} x_i \cdot 2^i \in \mathbb{Z}_{2^k} := \{0, \ldots, 2^k - 1\}$. Hence for any $l \in \{0, \ldots, k\}$ we have $(\widehat{v} \bmod 2^l) = \sum_{i=0}^{l-1} x_i \cdot 2^i \in \mathbb{Z}_{2^l}$, saying that the binary representation of $(\widehat{v} \bmod 2^l)$ consists of the l **lower bits** $(v \bmod 2^l) := [x_{l-1}, \ldots, x_0] \in \mathbb{Z}_2^l$ of v. Similarly, we have $(\widehat{v} \div 2^l) := \lfloor \frac{\widehat{v}}{2^l} \rfloor = \sum_{i=l}^{k-1} x_i \cdot 2^{i-l} = \sum_{i=0}^{k-l-1} x_{i+l} \cdot 2^i \in \mathbb{Z}_{2^{k-l}}$, saying that the binary representation of $(\widehat{v} \bmod 2^l)$ consists of the $k - l$ **upper bits** $(v \div 2^l) := [x_{k-1}, \ldots, x_l] \in \mathbb{Z}_2^{l-k}$ of v. Finally, for any $l \in \mathbb{N}_0$ we have $\widehat{v} \cdot 2^l = \sum_{i=0}^{k-1} x_i \cdot 2^{i+l} \in \mathbb{Z}_{2^{k+l}}$, saying that the binary representation of $\widehat{v} \cdot 2^l$ is given by $(v \cdot 2^l) := [x_{k-1}, \ldots, x_0, 0, \ldots, 0] \in \mathbb{Z}_2^{k+l}$, obtained by an l-fold **left bit shift** from v. Note that cutting out bits and bit shifts are computationally fast operations.

Now, in **output feedback mode (OFB)** an auxiliary block cipher of block length k is used. We choose a key $\pi \in \mathcal{S}_{\mathbb{Z}_2^k}$ and an initialisation vector $\widetilde{z}_0 \in \mathbb{Z}_2^k$, giving rise to a **keystream** $[\widetilde{z}_1, \widetilde{z}_2 \ldots] \subseteq \mathbb{Z}_2^k$. Genuine encryption takes place on blocks of length $l \leq k$, where for a given sequence of plaintext blocks $[v_1, v_2, \ldots] \subseteq \mathbb{Z}_2^l$ we recursively obtain the associated sequence of ciphertext blocks $[w_1, w_2, \ldots] \subseteq \mathbb{Z}_2^l$ as follows:

For $i \in \mathbb{N}$ let $\widetilde{z}_i := E_\pi(\widetilde{z}_{i-1}) \in \mathbb{Z}_2^k$, and then let $w_i := v_i \oplus (\widetilde{z}_i \div 2^{k-l}) \in \mathbb{Z}_2^l$. To decrypt, for $i \in \mathbb{N}$ let $\widetilde{z}_i := E_\pi(\widetilde{z}_{i-1}) \in \mathbb{Z}_2^k$, and then let $v_i := w_i \oplus (\widetilde{z}_i \div 2^{k-l}) \in \mathbb{Z}_2^l$. Indeed, noting that the sequence $[\widetilde{z}_1, \widetilde{z}_2, \ldots]$ is independent of ciphertext blocks, for $i \in \mathbb{N}$ we get $w_i \oplus (\widetilde{z}_i \div 2^{k-l}) = v_i \oplus (\widetilde{z}_i \div 2^{k-l}) \oplus (\widetilde{z}_i \div 2^{k-l}) = v_i$.

In particular, for $k = l$, we have an initialisation vector $z_0 \in \mathbb{Z}_2^l$, giving rise to the keystream $[z_1, z_2 \ldots]$ defined by $z_i := E_\pi(z_{i-1})$ for $i \in \mathbb{N}$, and thus encryption reads $v_i := w_i \oplus z_i$, and decryption reads $w_i := v_i \oplus z_i$, for $i \in \mathbb{N}$.

Both encryption and decryption need `xor` operations, cutting out bits and bit shifts, and an application of a permutation, where the former bit manipulations computationally are fast operations, while the latter is expensive. Moreover, since $\widetilde{z}_i = E_\pi(\widetilde{z}_{i-1})$ is independent of ciphertext blocks, the expensive part of decryption can be computed independently from any encryption going on. But ciphertext blocks only depend on their position in the sequence of blocks, but not on earlier ciphertext blocks, possibly making them easy to manipulate.

For example, letting $k := 4$, using the key $\pi := (1, 2, 3, 4) \in \mathcal{S}_4$ in a permutation cipher, and using the initialisation vector $\widetilde{z}_0 := [1, 0, 1, 0]$, the plaintext word `000 101 001 010`, grouped into blocks of length $l := 3$, yields the ciphertext word `010 000 011 111`, encryption and decryption being given as follows:

i	$\widetilde{z}_i$	$\widetilde{z}_i \div 2$	v_i	w_i
0	$[1, 0, 1, 0]$			
1	$[0, 1, 0, 1]$	$[0, 1, 0]$	$[0, 0, 0]$	$[0, 1, 0]$
2	$[1, 0, 1, 0]$	$[1, 0, 1]$	$[1, 0, 1]$	$[0, 0, 0]$
3	$[0, 1, 0, 1]$	$[0, 1, 0]$	$[0, 0, 1]$	$[0, 1, 1]$
4	$[1, 0, 1, 0]$	$[1, 0, 1]$	$[0, 1, 0]$	$[1, 1, 1]$

Letting $l := 4$ instead, the plaintext word `0001 0100 1010` yields the ciphertext word `0100 1110 1111`, where encryption and decryption are given as follows:

i	z_i	v_i	w_i
0	$[1, 0, 1, 0]$		
1	$[0, 1, 0, 1]$	$[0, 0, 0, 1]$	$[0, 1, 0, 0]$
2	$[1, 0, 1, 0]$	$[0, 1, 0, 0]$	$[1, 1, 1, 0]$
3	$[0, 1, 0, 1]$	$[1, 0, 1, 0]$	$[1, 1, 1, 1]$

d) In **cipher feedback mode (CFB)** we again consider an auxiliary block cipher of block length k. We choose a key $\pi \in \mathcal{S}_{\mathbb{Z}_2^k}$ and an initialisation vector $\widetilde{z}_0 \in \mathbb{Z}_2^k$, now giving rise to an **asynchronous keystream** $[\widetilde{z}_1, \widetilde{z}_2 \ldots] \subseteq \mathbb{Z}_2^k$, which also depends on genuine ciphertexts. Genuine encryption then takes place on blocks of length $l \leq k$, where for a given sequence of plaintext blocks $[v_1, v_2, \ldots] \subseteq \mathbb{Z}_2^l$ we recursively obtain the associated sequence of ciphertext blocks $[w_1, w_2, \ldots] \subseteq \mathbb{Z}_2^l$ as follows:

For $i \in \mathbb{N}$ let $w_i := v_i \oplus \big(E_\pi(\widetilde{z}_{i-1}) \div 2^{k-l}\big) \in \mathbb{Z}_2^l$ and $\widetilde{z}_i := \big((\widetilde{z}_{i-1} \bmod 2^{k-l}) \cdot 2^l\big) \oplus w_i \in \mathbb{Z}_2^k$; note that $\widetilde{z}_i$ is just the **concatenation** of $(\widetilde{v}_{i-1} \bmod 2^{k-l}) \in \mathbb{Z}_2^{k-l}$ and $w_i \in \mathbb{Z}_2^l$. To decrypt, for $i \in \mathbb{N}$ let $v_i := w_i \oplus \big(E_\pi(\widetilde{z}_{i-1}) \div 2^{k-l}\big) \in \mathbb{Z}_2^l$ and $\widetilde{z}_i := \big((\widetilde{v}_{i-1} \bmod 2^{k-l}) \cdot 2^l\big) \oplus w_i \in \mathbb{Z}_2^k$. Indeed, noting that the sequence $[\widetilde{z}_1, \widetilde{z}_2, \ldots]$ only depends on the ciphertext blocks $[w_1, w_2, \ldots]$, and thus is the same for the encryption and decryption processes, for $i \in \mathbb{N}$ we get $w_i \oplus \big(E_\pi(\widetilde{z}_{i-1}) \div 2^{k-l}\big) = v_i \oplus \big(E_\pi(\widetilde{z}_{i-1}) \div 2^{k-l}\big) \oplus \big(E_\pi(\widetilde{z}_{i-1}) \div 2^{k-l}\big) = v_i$.

In particular, for $k = l$ we get $\widetilde{z}_i = w_i$ for $i \in \mathbb{N}$, hence in particular we just have an initialisation vector $w_0 \in \mathbb{Z}_2^l$, and thus encryption reads $w_i := v_i \oplus E_\pi(w_{i-1})$, and decryption reads $v_i := w_i \oplus E_\pi(w_{i-1})$, for $i \in \mathbb{N}$.

Hence still both encryption and decryption need `xor` operations, cutting out bits and bit shifts, and an application of a permutation, where the former bit manipulations computationally are fast operations, while the latter is expensive. But $\widetilde{z}_{i-1}$ only depends on w_{i-1}, thus $E_\pi(\widetilde{z}_{i-1})$ can be computed without knowing the ciphertext block w_i to be decrypted, hence the expensive part of decryption can already be done while encryption of v_i still is under way. Moreover, the smaller $l \leq k$ is, the more often encryption and decryption have to be used, but the faster single blocks are transmitted, thus the choice of l is a trade-off between the encryption and decyption speed, that is essentially the application of a permutation, and transmission capacity.

For example, still letting $k := 4$, using the key $\pi := (1, 2, 3, 4) \in \mathcal{S}_4$ in a permutation cipher, and using the initialisation vector $\widetilde{v}_0 := [1, 0, 1, 0]$, the plaintext word `000 101 001 010`, grouped into blocks of length $l := 3$, yields the ciphertext word `010 101 100 001`, encryption and decryption being given as follows:

i	$\widetilde{v}_i$	$E_\pi(\widetilde{v}_{i-1})$	$\cdot \div 2$	v_i	w_i
0	$[1, 0, 1, 0]$				
1	$[0, 0, 1, 0]$	$[0, 1, 0, 1]$	$[0, 1, 0]$	$[0, 0, 0]$	$[0, 1, 0]$
2	$[0, 1, 0, 1]$	$[0, 0, 0, 1]$	$[0, 0, 0]$	$[1, 0, 1]$	$[1, 0, 1]$
3	$[1, 1, 0, 0]$	$[1, 0, 1, 0]$	$[1, 0, 1]$	$[0, 0, 1]$	$[1, 0, 0]$
4	$[0, 0, 0, 1]$	$[0, 1, 1, 0]$	$[0, 1, 1]$	$[0, 1, 0]$	$[0, 0, 1]$

Letting $l := 4$ instead, the plaintext word `0001 0100 1010` yields the ciphertext word `0100 0110 1001`, where encryption and decryption are given as follows:

i	$E_\pi(w_{i-1})$	v_i	w_i
0			$[1,0,1,0]$
1	$[0,1,0,1]$	$[0,0,0,1]$	$[0,1,0,0]$
2	$[0,0,1,0]$	$[0,1,0,0]$	$[0,1,1,0]$
3	$[0,0,1,1]$	$[1,0,1,0]$	$[1,0,0,1]$

(4.4) Remark. Block ciphers play a major role in practice, where actually **iterated block ciphers** are used:

a) Being given a block cipher over the alphabet $\mathcal{X} := \mathbb{Z}_2 = \{0,1\}$ of block length $l \in \mathbb{N}$, we obtain a **Feistel cipher** of **round number** $r \in \mathbb{N}$ and block length $2l$ as follows: Depending on the key of the Feistel cipher chosen, let $[e_1, \ldots, e_r]$ be a **key schedule**, that is a sequence of r keys of the underlying block cipher. To encrypt, a plaintext in $\mathcal{X}^{2l}$ is viewed as a concatenation $[v_0, w_0] \in (\mathcal{X}^l)^2$, allowing for recursively letting $[v_i, w_i] := [w_{i-1}, v_{i-1} \oplus E_{e_i}(w_{i-1})] \in (\mathcal{X}^l)^2$, for $i \in \{1, \ldots, r\}$, where the associated ciphertext is $[v_r, w_r]$. Hence, given $[v_r, w_r]$, decryption is performed by letting backwardly $[v_{i-1}, w_{i-1}] := [w_i \oplus E_{e_i}(v_i), v_i]$; note that the decryption functions of the underlying block cipher are not needed.

The **Data Encryption Standard (DES)** [1977] uses a Feistel cipher of block length 64 and round number 16, where the set of keys has cardinality $2^{56} \sim 7 \cdot 10^{16}$. We spare the details of how the key schedule is determined from the chosen key of the Feistel cipher, and how the encryption functions of the underlying block cipher are actually defined; see [12, Ch.3.5] and [2, Ch.5], We just note that, in contrast to the rounds of the Feistel cipher, and apart from the linear process of adding a round key, the encryption functions are non-linear, thought of being close to random permutations.

Various kinds of attacks against DES are known: Apart from the known-plaintext **linear cryptanalysis**, see [12, Ch.3.3], and the chosen-plaintext **differential cryptanalysis** [Biham-Shamir, 1991], see [12, Ch.3.4], the most successful one is the exhaustive search attack using the specially tailored 'DES Cracker' machine, which using distributed computing allows to break DES in a few hours. Still, DES continues to be useful as soon as multiple encryption is used. This is based on the fact the permutations induced by the 2^{56} keys do not form a subgroup of $\mathcal{S}_{2^{64}}$, but the subgroup thus generated is reported to have order at least 10^{2499}.

b) The **Advanced Encryption Standard (AES)** [2001] uses the so-called **Rijndael cipher** over the alphabet $\mathcal{X} := \mathbb{Z}_2 = \{0,1\}$ of block length 128, where various sets of keys of cardinality $2^{128} \sim 3 \cdot 10^{38}$ and $2^{192} \sim 3 \cdot 10^{57}$ and $2^{256} \sim 10^{77}$ are allowed, and where depending on the latter the round number equals 10 and 12 and 14, respectively.

Again we spare the details of how the key schedule is determined from the key chosen, and how the encryption functions of the underlying block cipher are actually defined; see [12, Ch.3.6] and [2, Ch.6]. But we note that, in contrast to the underlying block cipher of DES, the encryption functions of AES are more algebraic. More precisely, a plaintext in $\mathcal{X}^{128}$ is viewed as a concatenation in $(\mathcal{X}^8)^{16}$, that is a sequence of length 16 consisting of 8-bit sequences. Hence AES can be seen as being based on a block cipher of block length 8, and thus from the practical side is Byte-oriented. Moreover, $\mathcal{X}^8$ is identified with the finite field $\mathbb{F}_{256}$ of order $2^8 = 256$, given by $\mathbb{F}_{256} \cong \mathbb{Z}_2[T]/(T^8 + T^4 + T^3 + T + 1)$. Hence, apart from the linear process of adding a round key, the non-linear pieces of encryption are partly described in terms of the arithmetic of $\mathbb{F}_{256}$.

5 Affine ciphers

(5.1) Determinants and affine maps. a) Let R be a commutative ring. For $n \in \mathbb{N}$ let R^n be the set of n-tuples of elements of R, written as rows. Let $R^{n \times n}$ be the ring of $(n \times n)$-matrices over R, with usual matrix addition and multiplication and usual scalar multiplication.

For $A = [a_{ij}]_{ij} \in R^{n \times n}$ let $\det(A) := \sum_{\pi \in \mathcal{S}_n} \operatorname{sgn}(\pi) \cdot \prod_{i=1}^n a_{i,i\pi} \in R$ be the **determinant** of A, where $\operatorname{sgn}\colon \mathcal{S}_n \to \{\pm 1\}$ is the sign homomorphism. The determinant map is R-multilinear and alternating, that $\det(E_n) = 1$, where $E_n \in R^{n \times n}$ is the **identity matrix**, and that the row and column expansion formulae and the product rule hold.

For $i, j \in \{1, \ldots, n\}$ let $A_{ij} \in R^{(n-1)\times(n-1)}$ be the matrix obtained from A by deleting the i-th row and the j-th column, where for $n = 1$ we let $A_{11} := [] \in R^{0 \times 0}$. Then $\det(A_{ij}) \in R$ is called the $[i, j]$-th **minor** of A, where for $n = 1$ we let $\det(A_{11}) = \det([]) := 1 \in R$. Let $\operatorname{adj}(A) := [(-1)^{i+j} \cdot \det(A_{ji})]_{ij} \in R^{n \times n}$ be the **adjoint matrix** associated with A. Then we have the **adjointness theorem** $A \cdot \operatorname{adj}(A) = \operatorname{adj}(A) \cdot A = \det(A) \cdot E_n$:

For $i \in \{1, \ldots, n\}$ let $a_i = [a_{i1}, \ldots, a_{in}] \in R^n$ be the i-th row of A, and let $e_i \in R^n$ be the i-th **unit vector**. Thus for $i, k \in \{1, \ldots, n\}$ we have $(A \cdot \operatorname{adj}(A))_{ik} = \sum_{j=1}^n a_{ij} \cdot (-1)^{j+k} \cdot \det(A_{kj}) = \sum_{j=1}^n a_{ij} \cdot \det(a_1, \ldots, a_{k-1}, e_j, a_{k+1}, \ldots, a_n) = \det(a_1, \ldots, a_{k-1}, \sum_{j=1}^n a_{ij} e_j, a_{k+1}, \ldots, a_n) = \det(a_1, \ldots, a_{k-1}, a_i, a_{k+1}, \ldots, a_n)$. Thus $(A \cdot \operatorname{adj}(A))_{ik} = \det(A) \cdot \delta_{ik}$, showing $A \cdot \operatorname{adj}(A) = \det(A) \cdot E_n$. From $A^{\text{tr}} \cdot \operatorname{adj}(A^{\text{tr}}) = \det(A^{\text{tr}}) \cdot E_n$, using $\det(A^{\text{tr}}) = \det(A)$ and $\operatorname{adj}(A^{\text{tr}}) = \operatorname{adj}(A)^{\text{tr}}$, we also get $\operatorname{adj}(A) \cdot A = \det(A) \cdot E_n$. ♯

Thus $A \in R^{n \times n}$ is **invertible**, i. e. there are $B, B' \in R^{n \times n}$ such that $AB = B'A = E_n$, if and only if $\det(A) \in R^*$, where R^* is the group of units of R: From $AB = E_n$ we get $\det(A)\det(B) = \det(AB) = \det(E_n) = 1$, hence $\det(A) \in R^*$, and the converse follows from the adjointness theorem. The set of invertible $(n \times n)$-matrices over R is a multiplicative group $\mathrm{GL}_n(R) = (R^{n \times n})^*$, being called the **general linear group** of size n over R; in particular we have $\mathrm{GL}_1(R) = R^*$. The **inverse** $A^{-1} = B = B' \in \mathrm{GL}_n(R)$ of $A \in \mathrm{GL}_n(R)$ is unique and given as $A^{-1} = \det(A)^{-1} \cdot \operatorname{adj}(A)$.

b) Given $A \in R^{n \times n}$ and $b \in R^n$, the map $\varphi_{A,b}\colon R^n \to R^n\colon v \mapsto vA + b$ is called an **affine R-linear map**; for $b = 0$ we get the **R-linear map** $\varphi_{A,0} = \varphi_A \colon R^n \to R^n \colon v \mapsto vA$. The map $\varphi_{A,b} \colon R^n \to R^n$ is a bijection if and only if $A \in \mathrm{GL}_n(R)$; in this case its inverse is given as $\varphi_{A^{-1},-bA^{-1}} \colon R^n \to R^n$, thus also is affine R-linear:

If $A \in \mathrm{GL}_n(R)$ then $v\varphi_{A,b}\varphi_{A^{-1},-bA^{-1}} = (vA+b)\varphi_{A^{-1},-bA^{-1}} = (vA+b)A^{-1} - bA^{-1} = v$ and $v\varphi_{A^{-1},-bA^{-1}}\varphi_{A,b} = (vA^{-1} - bA^{-1})\varphi_{A,b} = (vA^{-1} - bA^{-1})A + b = v$, for all $v \in R^n$. Conversely, if $\varphi_{A,b}$ is a bijection, then since by the above $\varphi_{E_n,-b}$ is a bijection, the R-linear map $\varphi_A = \varphi_{A,b}\varphi_{E_n,-b}$ is a bijection as well, and since φ_A is R-linear we have $A \in \mathrm{GL}_n(R)$ and $\varphi_A^{-1} = \varphi_{A^{-1}}$. ♯

(5.2) Affine ciphers. Let $n \in \mathbb{N}$ and $R := \mathbb{Z}_n$. A block cipher over R of block length $l \in \mathbb{N}$ is called an **affine cipher**, if the set of keys is given as $\mathcal{K} := \mathrm{GL}_l(R) \times R^l$, and the encryption and decryption functions associated with $[A, b] \in \mathcal{K}$ are the affine R-linear maps $E_{A,b} = \varphi_{A,b} \colon R^l \to R^l \colon v \mapsto vA + b$ and $D_{A,b} = \varphi_{A,b}^{-1} = \varphi_{A^{-1},-bA^{-1}} \colon R^l \to R^l \colon v \mapsto (v-b)A^{-1}$, respectively. Hence affine ciphers are symmetric cryptosystems.

For $A = E_l$ we in particular obtain the **Vigenère cipher** [1553], whose set of keys is $\mathcal{K} := R^l$, and whose encryption and decryption functions for $b \in R^l$ are the **translations** $E_b = \varphi_{E_l,b} \colon R^l \to R^l \colon v \mapsto v + b$ and $D_b = \varphi_{E_l,-b} \colon R^l \to R^l \colon v \mapsto v - b$, respectively. Hence $|\mathcal{K}| = |R|^l = n^l$; for $l = 1$ we recover the shift cipher.

For $b = 0$ we in particular obtain the **Hill cipher** [1929], whose set of keys is $\mathcal{K} := \mathrm{GL}_l(R)$, and whose encryption and decryption functions for $A \in \mathrm{GL}_l(R)$ are the R-linear maps $E_A = \varphi_{A,0} \colon R^l \to R^l \colon v \mapsto vA$ and $D_A = \varphi_{A^{-1},0} \colon R^l \to R^l \colon v \mapsto vA^{-1}$, respectively. Hence the Hill cipher is the most general **R-linear cipher**, and we have $|\mathcal{K}| = |\mathrm{GL}_l(R)|$; we recover the permutation cipher with key $\pi \in \mathcal{S}_l$ by using the **permutation matrix** $A_\pi := [\delta_{j,i\pi}]_{ij} \in R^{l \times l}$; we have $A_\pi^{-1} = A_{\pi^{-1}} = A_\pi^{\mathrm{tr}} \in \mathrm{GL}_l(R)$.

(5.3) Breaking affine ciphers. Ciphertext-only attacks against general affine ciphers might be difficult, but they are vulnerable to known-plaintext attacks:

We first consider a Hill cipher, and assume we have l plaintext-ciphertext pairs $[v_1, w_1], \ldots, [v_l, w_l] \in R^l \times R^l$, thus $w_i = v_i A \in R^l$, where $A \in \mathcal{K}$. Letting $M := [v_1, \ldots, v_l] \in R^{l \times l}$ and $W := [w_1, \ldots, w_l] \in R^{l \times l}$ be the matrices whose rows consist of the $v_i \in R^l$ and $w_i \in R^l$, respectively, we thus have $W = MA$. Additionally assuming that $M \in \mathrm{GL}_l(R)$ we get $A = M^{-1}W \in R^{l \times l}$; if M is not invertible more plaintext-ciphertext pairs must be collected.

In the case of an arbitrary affine cipher, assume we have $l+1$ plaintext-ciphertext pairs $[v_0, w_0], \ldots, [v_l, w_l] \in R^l \times R^l$. Then we have $w_i = v_i A + b \in R^l$, where $[A, b] \in \mathcal{K}$. Letting $M := [v_1 - v_0, \ldots, v_l - v_0] \in R^{l \times l}$ and $W := [w_1 - w_0, \ldots, w_l - w_0] \in R^{l \times l}$ be the matrices whose rows consist of $v_i - v_0 \in R^l$ and $w_i - w_0 \in R^l$, respectively, we thus have $W = MA$. Additionally assuming that $M \in \mathrm{GL}_l(R)$ we get $A = M^{-1}W \in R^{l \times l}$; if M is not invertible more plaintext-ciphertext

Table 4: Frequency of letters.

$\mathcal{X}_{\text{latin}}$	frequency
a	2
b	1
c	0
d	7
e	5
f	4
g	0
h	5
i	0
j	0
k	5
l	2
m	2

$\mathcal{X}_{\text{latin}}$	frequency
n	1
o	1
p	2
q	0
r	8
s	3
t	0
u	2
v	4
w	0
x	2
y	1
z	0

pairs must be collected. From knowing A we get $b = w_i - v_i A \in R^l$ anyway.

i) For example, we consider a Hill cipher over $R := \mathbb{Z}_{26}$ with block length $l = 3$, and we assume we know the following plaintext-ciphertext pair:

`conversation` ↦ `alspuplauung`

This yields $[v_1, \ldots, v_4] = [[18, 20, 1], [18, 19, 8], [19, 20, 19], [8, 14, 13]] \subseteq \mathbb{Z}_{26}^3$ and $[w_1, \ldots, w_4] = [[0, 11, 18], [15, 20, 15], [11, 0, 20], [20, 13, 6]] \subseteq \mathbb{Z}_{26}^3$. Letting $M := [v_1, \ldots, v_3] \in \mathbb{Z}_{26}^{3\times 3}$ and $W := [w_1, \ldots, w_3] \in \mathbb{Z}_{26}^{3\times 3}$ we get $\det(M) = 25 \in \mathbb{Z}_{26}^*$, hence $M \in \mathrm{GL}_3(\mathbb{Z}_{26})$, and thus

$$A := M^{-1}W = \begin{bmatrix} 7 & 22 & 15 \\ 8 & 15 & 22 \\ 1 & 6 & 18 \end{bmatrix} \cdot \begin{bmatrix} 0 & 11 & 18 \\ 15 & 20 & 15 \\ 11 & 0 & 20 \end{bmatrix} = \begin{bmatrix} 1 & 23 & 2 \\ 25 & 24 & 3 \\ 2 & 1 & 0 \end{bmatrix} \in \mathbb{Z}_{26}^{3\times 3},$$

where indeed $\det(A) = 11 \in \mathbb{Z}_{26}^*$, hence $A \in \mathrm{GL}_3(\mathbb{Z}_{26})$. Note that for $M' := [v_1, v_3, v_4] \in \mathbb{Z}_{26}^{3\times 3}$ we get $\det(M') = 22 \in \mathbb{Z}_{26} \setminus \mathbb{Z}_{26}^*$, implying that $M' \notin \mathrm{GL}_3(\mathbb{Z}_{26})$, so that we cannot use M' for this attack.

ii) For example, in the substitution cipher case $l = 1$, where hence $\mathcal{K} = R^* \times R$, this method even allows for a ciphertext-only attack. Let $R = \mathbb{Z}_{26}$ and consider the following ciphertext of length 57:

`fmxvedkaphferbndkrxrsrefmorudsdkdvshvufedkaprkdlyevlrhhrh`

We try to find the key $[a, b] \in \mathcal{K}$ used and to determine the plaintext, again using statistical properties of the plaintext language. The frequency of occurrence of the various letters is given in Table 4. The most frequent ciphertext letters are `r` and `{d, e, h, k}`. We conjecture that the most frequent plaintext letters are `e, t`, and hence that $E_{a,b}(\mathtt{e}) = \mathtt{r}$ and $E_{a,b}(\mathtt{t}) \in \{\mathtt{d},\ \mathtt{e},\ \mathtt{h},\ \mathtt{k}\}$. This yields $E_{a,b}(4) = 4a + b = 17 \in \mathbb{Z}_{26}$ and $E_{a,b}(19) = 19a + b \in \{3, 4, 7, 10\} \subseteq \mathbb{Z}_{26}$. Hence we have $15a \in \{12, 13, 16, 19\} \subseteq \mathbb{Z}_{26}$, and since indeed $15 \in \mathbb{Z}_{26}^*$ we conclude $a \in 15^{-1} \cdot \{12, 13, 16, 19\} = 7 \cdot \{12, 13, 16, 19\} = \{6, 13, 8, 3\} \subseteq \mathbb{Z}_{26}$. For these cases we only have $3 \in \mathbb{Z}_{26}^*$. Thus conjecturing $a = 3 \in \mathbb{Z}_{26}$ yields $b = 5 \in \mathbb{Z}_{26}$, and trying $D_{3,5} \colon \mathbb{Z}_{26} \to \mathbb{Z}_{26} \colon x \mapsto (x - 5) \cdot 3^{-1} = 9x + 7$ indeed yields:

```
algorithms are quite general definitions of arithmetic processes
```

(5.4) Breaking Vigenère ciphers. We show a ciphertext-only attack, where we first aim to determe the block length l. We present two variants to do this:

i) The **Kasiski-Babbage test** [1863, 1854] is based on the fact that plaintext words are encrypted to the same ciphertext words if their positions differ by a multiple of l. Hence we look for ciphertext words of a fixed not too small length occurring more than once, and conjecture that l divides the greatest common divisor of their mutual distances; this is feasible for arbitrary block lengths.

ii) We consider the **index of coincidence** [Friedman, 1920]: If letters are uniformly distributed, the probability that picking two letters uniformly and independently yields one and the same letter equals $\sum_{i \in \mathbb{Z}_{26}} \frac{1}{26^2} = \frac{1}{26} \sim 0.0385$; using the probability distribution $u := \frac{1}{26} \cdot [1, \ldots, 1] \in \mathbb{R}^{26}$, this can be rephrased as computing the standard scalar product $\langle u, u \rangle = \frac{1}{26}$. In the same vein, letting $p := [p_0, \ldots, p_{25}] \in \mathbb{R}^{26}$, where p_i, for $i \in \mathbb{Z}_{26}$, is the probability of occurrence of letters in plaintexts as given in Table 2, the above probability becomes $\langle p, p \rangle = \sum_{i \in \mathbb{Z}_{26}} p_i^2 = \frac{65601}{10^6} \sim 0.0656$; this number does not change under substitution. Note that, since $\langle u, p \rangle = \frac{1}{26} \cdot \sum_{i \in \mathbb{Z}_{26}} p_i = \frac{1}{26}$, the Cauchy-Schwarz inequality $\langle u, p \rangle^2 \leq \langle u, u \rangle \cdot \langle p, p \rangle$ implies $\langle p, p \rangle \geq \frac{1}{26}$, with equality if and only if $p = u$.

A word w of length $l(w)$ is interpreted as the outcome of picking $l(w)$ letters uniformly and independently. Letting $f_i(w) \in \{0, \ldots, l(w)\}$ be the frequency of occurrence of the letter $i \in \mathbb{Z}_{26}$, there are $\sum_{i \in \mathbb{Z}_{26}} \binom{f_i(w)}{2}$ unordered pairs of equal letters, amongst all $\binom{l(w)}{2}$ unordered pairs. Hence the **index of coincidence** $I(w) := \frac{1}{\binom{l(w)}{2}} \cdot \sum_{i \in \mathbb{Z}_{26}} \binom{f_i(w)}{2} = \frac{1}{l(w)(l(w)-1)} \cdot \sum_{i \in \mathbb{Z}_{26}} f_i(w)(f_i(w) - 1)$ approximates the above probability $\langle p, p \rangle$.

To apply this, let $v = [v_1, v_2, \ldots] \subseteq \mathbb{Z}_{26}$ be a ciphertext. Then for $k \in \mathbb{N}$ and $j \in \{1, \ldots, k\}$ we consider the word $w_j := [v_j, v_{j+k}, v_{j+2k}, \ldots] \subseteq \mathbb{Z}_{26}$. Hence if the block length l divides k, all letters in w_j are shifted by the same element $b_j \in \mathbb{Z}_{26}$, hence we expect $I(w_j) \sim \langle p, p \rangle \sim 0.0656$. If l does not divide k, then each letter in w_j can be considered to be shifted by a random element of $\mathbb{Z}_{26}$, thus the letters in w_j should be uniformly distributed, hence we expect $I(w_j) \sim \langle u, u \rangle \sim 0.0385$. The numbers $I(w_j)$ are computed successively for increasing values of $k \in \mathbb{N}$; hence this is feasible only for small block lengths.

iii) Having a conjecture for the block length l, we aim to find the key used, that is the translation $b = [b_1, \ldots, b_l] \in \mathbb{Z}_{26}^l$:

Table 5: Kasiski-Babbage test.

word		positions
tui	$[19, 20, 8]$	$167, 237, 277, 287$
chr	$[2, 7, 17]$	$1, 36$
voa	$[21, 14, 0]$	$6, 161$
atb	$[0, 19, 1]$	$14, 124$
mnn	$[12, 13, 13]$	$20, 55$
nng	$[13, 13, 6]$	$21, 76$
tce	$[19, 2, 4]$	$95, 220$
ceq	$[2, 4, 16]$	$96, 221$
tbo	$[19, 1, 14]$	$120, 125$
xan	$[23, 0, 13]$	$194, 224$
axa	$[0, 23, 0]$	$163, 179$

Let $\pi \in \mathcal{S}_{\mathbb{Z}_{26}}$ be any permutation, and let $p_\pi := [p_{0\pi^{-1}}, \ldots, p_{25\cdot\pi^{-1}}] \in \mathbb{R}^{26}$ be the accordingly substituted probability distribution. Then we have $\langle p_\pi, p_\pi \rangle = \langle p, p \rangle$, and hence the Cauchy-Schwarz inequality $\langle p, p_\pi \rangle^2 \leq \langle p, p \rangle \cdot \langle p_\pi, p_\pi \rangle$ implies $\sum_{i \in \mathbb{Z}_{26}} p_i p_{i\pi^{-1}} = \langle p, p_\pi \rangle \leq \langle p, p \rangle$, with equality if and only if $p_\pi = p$, that is π leaves the subsets of $\mathbb{Z}_{26}$ consisting of letters with equal probability fixed.

Now let p_m be the probability distribution obtained by the permutation induced by shifting with $m \in \mathbb{Z}_{26}$. Applying $m \neq 0$ can be considered as a random non-trivial shift, hence we expect that $\langle p, p_m \rangle$ is independent of the choice of $m \neq 0$. Thus from $\sum_{m \in \mathbb{Z}_{26}} \langle p, p_m \rangle = \langle p, \sum_{m \in \mathbb{Z}_{26}} p_m \rangle = \langle p, [1, \ldots, 1] \rangle = 1$ we get $\langle p, p_m \rangle \sim \frac{1}{25}(1 - \langle p, p \rangle) \sim 0.0374$.

As far as the permutation induced by shifting with b_j is concerned, we observe that the probability p_{i-b_j} is approximated by $\frac{f_i(w_j)}{l(w_j)}$, for $i \in \mathbb{Z}_{26}$. Thus varying $m \in \mathbb{Z}_{26}$, the sum $I_m(w_j) := \frac{1}{l(w_j)} \cdot \sum_{i \in \mathbb{Z}_{26}} p_i f_{i+m}(w_j)$ should be maximal for $m = b_j$, in which case we expect $I_m(w_j) \sim \langle p, p_{b_j - m} \rangle = \langle p, p \rangle \sim 0.0656$. If $m \neq b_j$, then the letters in w_j can be considered to be shifted by a random element in $\mathbb{Z}_{26} \setminus \{0\}$, hence we expect $I_m(w_j) \sim \frac{1}{25}(1 - \langle p, p \rangle) \sim 0.0374$. The numbers $I_m(w_j)$ are computed for all $m \in \mathbb{Z}_{26}$, which amounts to $26 \cdot l$ checks, instead of 26^l inecessary for an exhaustive search.

E. g. we consider the following ciphertext of length 314:

```
chreevoahmaeratbiaemnngemrvrfexsfsfchrhthsjik
nlbrznrbjmnnrrwrntabchzezwismvnngionnvrzbosgh
arhktceqtbwkfobnsglxquaxbwgfitboaatboiikfigla
xuahljnqknwscymjwncyxrfmqvoaxabtuiornrctadfax
aeoylhatebwafxanwrpexrtegrzrhyjrzikbteitceqxa
niezbwefegmtuifxrrptlknhtrsvgtunrmzqbbvlquevb
ndgswxtuiiauamgptuirbhbyemhnzxmoaibwnbzxvbrv
```

Table 6: Index of coincidence.

k	j	$I(w_j) \sim$
1	1	0.0463
2	1	0.0476
2	2	0.0482
3	1	0.0491
3	2	0.0447
3	3	0.0424
4	1	0.0458
4	2	0.0448
4	3	0.0500
4	4	0.0476
5	1	0.0742
5	2	0.0630
5	3	0.0742
5	4	0.0625
5	5	0.0635
6	1	0.0385
6	2	0.0493
6	3	0.0468
6	4	0.0649
6	5	0.0505
6	6	0.0483
7	1	0.0495
7	2	0.0505
7	3	0.0545
7	4	0.0515
7	5	0.0495
7	6	0.0444
7	7	0.0423
8	1	0.0538
8	2	0.0474
8	3	0.0661
8	4	0.0567
8	5	0.0445
8	6	0.0526
8	7	0.0567
8	8	0.0432
9	1	0.0706
9	2	0.0403
9	3	0.0319
9	4	0.0454
9	5	0.0319
9	6	0.0286
9	7	0.0437
9	8	0.0403
9	9	0.0321
10	1	0.0625
10	2	0.0726
10	3	0.0706
10	4	0.0484
10	5	0.0731
10	6	0.0774
10	7	0.0538
10	8	0.0710
10	9	0.0989
10	10	0.0559

We look for words of length 3 occurring more than once, and collect their positions as given in Table 5. From this we conjecture $l = 5$. Similarly, for $k \in \{1, \ldots, 6\}$ we compute the indices of coincidence $I(w_1), \ldots, I(w_k)$ as given in Table 6, and again we are led to the conjecture $l = 5$. Finally, for all w_j and $m \in \mathbb{Z}_{26}$ we compute $I_m(w_j)$ as given in Table 7. This yields the conjecture $b = [9, 0, 13, 4, 19] \in \mathbb{Z}_{26}^{6}$ which decodes to `janet`. Indeed we get:

```
the almond tree was in a tentative blossom the days wer
e longer often ending with magnificent evenings of co
rrugated pink skies the hunting season was over with h
ounds and guns put away for six months the vineyards we
re busy again as the well organized farmers treated th
eir vines and the more lackadaisical neighbors hurri
ed to do the pruning they should have done in november
```

Table 7: Shifts.

j	$I_m(w_j) \sim$								
1	0.0338	0.0304	0.0347	0.0365	0.0364	0.0429	0.0301	0.0303	0.0464
	$\boxed{0.0716}$	0.0376	0.0267	0.0306	0.0493	0.0369	0.0370	0.0311	0.0309
	0.0350	0.0458	0.0465	0.0371	0.0429	0.0404	0.0415	0.0372	
2	$\boxed{0.0623}$	0.0394	0.0316	0.0414	0.0388	0.0352	0.0444	0.0370	0.0291
	0.0409	0.0435	0.0363	0.0323	0.0479	0.0432	0.0412	0.0376	0.0350
	0.0305	0.0352	0.0388	0.0365	0.0417	0.0293	0.0249	0.0457	
3	0.0476	0.0426	0.0407	0.0361	0.0327	0.0338	0.0372	0.0327	0.0327
	0.0441	0.0308	0.0269	0.0481	$\boxed{0.0696}$	0.0430	0.0329	0.0323	0.0427
	0.0353	0.0361	0.0308	0.0287	0.0348	0.0410	0.0462	0.0401	
4	0.0509	0.0344	0.0312	0.0340	$\boxed{0.0630}$	0.0339	0.0370	0.0376	0.0435
	0.0267	0.0330	0.0372	0.0362	0.0355	0.0347	0.0443	0.0407	0.0456
	0.0306	0.0444	0.0424	0.0417	0.0377	0.0382	0.0330	0.0323	
5	0.0443	0.0402	0.0320	0.0269	0.0362	0.0402	0.0448	0.0360	0.0505
	0.0335	0.0341	0.0380	0.0451	0.0383	0.0335	0.0429	0.0333	0.0331
	0.0365	$\boxed{0.0607}$	0.0348	0.0345	0.0327	0.0501	0.0333	0.0345	

6 Stream ciphers

(6.1) Stream ciphers. A symmetric cryptosystem $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ together with a **keystream generator** $\mathcal{S} = \{S_i; i \in \mathbb{N}\}$, consisting of maps $S_i \colon \mathcal{K} \times \mathcal{C}^{i-1} \to \mathcal{K}$, is called a **stream cipher**. A stream cipher is called **synchronous**, if the maps S_i only depend on $\mathcal{K}$, otherwise it is called **asynchronous**.

In the synchronous case, for a **seed** key $k_0 \in \mathcal{K}$ let $k_i := S_i(k_0) \in \mathcal{K}$, for $i \in \mathbb{N}$, be the associated **keystream**. A keystream is called **periodic** of period $l \in \mathbb{N}$, if for all seed keys $k_0 \in \mathcal{K}$ we have $k_{i+l} = k_i \in \mathcal{K}$, for $i \in \mathbb{N}_0$.

The idea is, starting with a seed $k_0 \in \mathcal{K}$, to vary the key $S_i(k_0, [p_1, \ldots, p_{i-1}]) \in \mathcal{K}$ used for each plaintext $p_i \in \mathcal{P}$ encrypted, possibly depending on the earlier plaintexts $[p_1, \ldots, p_{i-1}] \in \mathcal{P}^{i-1}$. The idea of synchronous ciphers is that Alice and Bob are able to compute the keystream simultaneously, while for asynchronous ciphers Alice has to receive and decrypt earlier ciphertexts first.

E. g. any conventional cryptosystem can be considered as a synchronous stream cipher with constant keystream $k_i = k_0 \in \mathcal{K}$, for $i \in \mathbb{N}$. E. g. the operation modes ECB and OFB of block ciphers over $\mathbb{Z}_2$ are synchronous stream ciphers, where the former has a constant keystream, while CBC and CFB are asynchronous stream ciphers. E. g. the Vigenère cipher over R of block length $l \in \mathbb{N}$ can be considered as a synchronous stream cipher of period $l \in \mathbb{N}$, where $\mathcal{P} = \mathcal{C} = R$: For a key $b = [b_1, \ldots, b_l] \in R^l$ let $k_i = b_i$ for $i \in \{1, \ldots, l\}$, and $k_i := k_{i-l}$ for $i \geq l + 1$. This indicates that synchronous stream ciphers of short periods are vulnerable to ciphertext-only attacks using the Kasiski-Babbage test or coincidence indices.

(6.2) Autokey ciphers. The **autokey cipher** is the asynchronous stream cipher given as follows: Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_n$, where $n \in \mathbb{N}$, and for $k \in \mathbb{Z}_n$ let $E_k \colon \mathbb{Z}_n \to \mathbb{Z}_n \colon x \mapsto x + k$ and $D_k \colon \mathbb{Z}_n \to \mathbb{Z}_n \colon x \mapsto x - k$ be the encryption and decryption functions of the shift cipher. For a seed $k_0 \in \mathbb{Z}_n$, the keystream generator is defined as $k_1 = S_1(k_0) := k_0 \in \mathbb{Z}_n$, and $k_i = S_i(k_0, [c_1, \ldots, c_{i-1}]) := D_{k_{i-1}}(c_{i-1}) = p_{i-1} \in \mathbb{Z}_n$ for $i \geq 2$,

E. g. for $n = 26$ the plaintext `rendezvous` gives $[17, 4, 13, 3, 4, 25, 21, 14, 20, 18]$, and the keystream associated with $k_0 = 8$ is $[8, 17, 4, 13, 3, 4, 25, 21, 14, 20]$. This yields $[25, 21, 17, 16, 7, 3, 20, 9, 8, 12]$, thus the ciphertext is `zvrqhdujim`.

(6.3) LFSR-based stream ciphers. a) In practice, stream ciphers are often realized over the finite field $\mathbb{F}_2 = \mathbb{Z}_2$. Hence more generally we let $\mathcal{P} = \mathcal{C} = \mathcal{K} := \mathbb{F}_q$ be the finite field with q elements; thus q might be any prime power, but typically we consider the case $\mathbb{F}_p = \mathbb{Z}_p$, where p is a prime. For $k \in \mathbb{F}_q$ let again $E_k \colon \mathbb{F}_q \to \mathbb{F}_q \colon x \mapsto x + k$ and $D_k \colon \mathbb{F}_q \to \mathbb{F}_q \colon x \mapsto x - k$. This becomes a periodic synchronous stream cipher by using a **linear recurrence** of **degree** $d \in \mathbb{N}$; in practice these are realized by **linear feedback shift registers (LFSR)**:

Let $\gamma := [c_0, \ldots, c_{d-1}] \in \mathbb{F}_q^d$ be fixed. For a seed $\kappa := [k_1, \ldots, k_d] \in \mathbb{F}_q^d$ let the keystream be defined by $k_i := \sum_{j=0}^{d-1} c_j k_{i-d+j} = [k_{i-d}, \ldots, k_{i-1}] \cdot \gamma^{\mathrm{tr}} \in \mathbb{F}_q$ for $i \geq d + 1$. Hence we may assume that $c_0 \neq 0 \in \mathbb{F}_q$, since otherwise we have a linear recurrence of degree $d - 1$. For the seed $\kappa = 0 \in \mathbb{F}_q^d$ we get $k_i = 0$ for all $i \in \mathbb{N}$, and thus the identity as encryption and decryption functions, hence the interesting case is $\kappa \neq 0$.

For $i \geq d + 1$ we have $[k_{i-d+1}, \ldots, k_i] = [k_{i-d}, \ldots, k_{i-1}] \cdot C_\gamma \in \mathbb{F}_q^d$, where

$$C_\gamma := \begin{bmatrix} 0 & & & & \cdots & c_0 \\ 1 & 0 & & & \cdots & c_1 \\ 0 & 1 & 0 & & \cdots & c_2 \\ \vdots & & \ddots & \ddots & & \vdots \\ & & & 1 & 0 & c_{d-2} \\ 0 & \cdots & & & 1 & c_{d-1} \end{bmatrix} \in \mathbb{F}_q^{d \times d}$$

is the (transposed of the) **companion matrix** of the **generating polynomial** $p_\gamma := X^d - \sum_{j=0}^{d-1} c_j X^j \in \mathbb{F}_q[X]$. Hence we have $\det(C_\gamma) = (-1)^{d-1} \cdot c_0 \in \mathbb{F}_q$, thus the assumption $c_0 \neq 0$ amounts to saying that C_γ is invertible, and hence the associated $\mathbb{F}_q$-linear map $\mathbb{F}_q^d \to \mathbb{F}_q^d \colon \kappa \mapsto \kappa \cdot C_\gamma$ is bijective. Thus for any $\kappa \in \mathbb{F}_q^d$ the keystream is periodic, the seed $\kappa = 0$ being a fixed point; note that since $|\mathbb{F}_q^d| = q^d$ is finite, in the case $c_0 = 0$ the keystream still is finally periodic.

b) To determine periods, we first determine the **characteristic** and **minimum polynomials** of C_γ: For the **unit vector** $e_1 := [1, 0, \ldots, 0] \in \mathbb{F}_q^d$, the sequence $[C_\gamma^i \cdot e_1^{\mathrm{tr}}; i \in \{0, \ldots, d-1\}] = [e_1^{\mathrm{tr}}, e_2^{\mathrm{tr}}, \ldots, e_d^{\mathrm{tr}}] \subseteq \mathbb{F}_q^{d \times 1}$ is $\mathbb{F}_q$-linearly independent, while $C_\gamma^d \cdot e_1^{\mathrm{tr}} = C_\gamma \cdot e_d^{\mathrm{tr}} = \sum_{j=1}^d c_{j-1} e_j^{\mathrm{tr}} \in \mathbb{F}_q^{d\times 1}$. This implies that the minimum polynomial of $e_1^{\mathrm{tr}} \in \mathbb{F}_q^{d\times 1}$ with respect to C_γ equals $\mu_{C_\gamma, e_1^{\mathrm{tr}}} = p_\gamma \in \mathbb{F}_q[X]$. Since $\mu_{C_\gamma, e_1^{\mathrm{tr}}}$ divides the minimum polynomial $\mu_{C_\gamma} \in \mathbb{F}_q[X]$ of C_γ, and by the Cayley-Hamilton Theorem the latter divides the characteristic polynomial $\chi_{C_\gamma} := \det(X \cdot E_d - C_\gamma) \in \mathbb{F}_q[X]$, which in turn has degree d, we conclude that $\chi_{C_\gamma} = \mu_{C_\gamma} = \mu_{C_\gamma, e_1^{\mathrm{tr}}} = p_\gamma \in \mathbb{F}_q[X]$.

Thus for the minimum polynomial $\mu_{C_\gamma,\kappa} \in \mathbb{F}_q[X]$ of $\kappa \in \mathbb{F}_q^d$ with respect to C_γ we have $\mu_{C_\gamma,\kappa} \mid p_\gamma \in \mathbb{F}_q[X]$. Hence we may assume that $\mu_{C_\gamma,\kappa} = p_\gamma$, otherwise we use $\mu_{C_\gamma,\kappa}$ instead as a generating polynomial of smaller degree; if p_γ is irreducible, then for any $\kappa \neq 0$ we have $\mu_{C_\gamma,\kappa} = p_\gamma$ anyway. Now the period of κ is given as $l \in \mathbb{N}$ minimal such that $\kappa \cdot C_\gamma^l = \kappa$, that is $\kappa \cdot (C_\gamma^l - E_d) = 0$, thus the period of κ is given as $l \in \mathbb{N}$ minimal such that $p_\gamma \mid X^l - 1 \in \mathbb{F}_q[X]$.

Since $|\mathbb{F}_q^d \setminus \{0\}| = q^d - 1$, the best possible case is $l = q^d - 1$, in other words all $\kappa \neq 0$ are in one and the same C_γ-orbit. Actually, there always is an irreducible polynomial p_γ of degree d yielding period $q^d - 1$:

The finite field $\mathbb{F}_{q^d}$ can be considered as an $\mathbb{F}_q$-vector space of $\mathbb{F}_q$-dimension d. Thus for any $a \in \mathbb{F}_{q^d}$ we have an associated $\mathbb{F}_q$-linear map $\widehat{a}\colon \mathbb{F}_{q^d} \to \mathbb{F}_{q^d} : x \mapsto xa$. Then $\mu_a := \mu_{\widehat{a}} \in \mathbb{F}_q[X]$ is called the **minimum polynomial** of $a \in \mathbb{F}_{q^d}$ over $\mathbb{F}_q$. Hence it follows from $\mathbb{F}_{q^d}$ being an integral domain that μ_a is irreducible. Moreover, if $\mu_a \mid X^l - 1$, then we have $a^l = 1 \in \mathbb{F}_{q^d}^*$. Now by **Artin's Theorem**, see (8.3), there is a **primitive root** $\rho \in \mathbb{F}_{q^d}^*$, that is $l = q^d - 1$ is minimal such that $\rho^l = 1$, implying that $\mu_\rho \in \mathbb{F}_q[X]$ has the desired properties. ♯

c) This leads to the following known-plaintext attack, aiming at finding the degree d and the vector $\gamma \in \mathbb{F}_q^d$: Let $[v_i, w_i] \in \mathbb{F}_q \times \mathbb{F}_q$ for $i \in \{1, \ldots, 2n\}$ be plaintext-ciphertext pairs, where $n \in \mathbb{N}$. Then we have $k_i = w_i - v_i \in \mathbb{F}_q$, and we let

$$M_n := \begin{bmatrix} k_1 & k_2 & \cdots & k_n \\ k_2 & k_3 & \cdots & k_{n+1} \\ \vdots & \vdots & & \vdots \\ k_n & k_{n+1} & \cdots & k_{2n-1} \end{bmatrix} \in \mathbb{F}_q^{n\times n}.$$

For $n \geq d = \deg(p_\gamma)$ there is a matrix $P_{n,d} = [E_d; \gamma, *, \ldots, *] \in \mathbb{F}_q^{n\times d}$, for suitable rows '$*$' in $\mathbb{F}_q^d$, such that $M_n = P_{n,d} \cdot M_d \cdot P_{n,d}^{\mathrm{tr}}$, implying that $\mathrm{rk}(M_n) = \mathrm{rk}(M_d)$. Moreover, we have $M_d = [\kappa, \kappa \cdot C_\gamma, \ldots, \kappa \cdot C_\gamma^{d-1}] \in \mathbb{F}_q^{d\times d}$, hence from $p_\gamma = \mu_{C_\gamma,\kappa}$ we infer that $\mathrm{rk}(M_d) = d$. Thus we find d conjecturally by computing $\mathrm{rk}(M_n)$ for increasing values of $n \in \mathbb{N}$. Finally, we have $\gamma \cdot M_d = [k_{d+1}, \ldots, k_{2d}] \in \mathbb{F}_q^d$, and since M_d is invertible we get $\gamma = [k_{d+1}, \ldots, k_{2d}] \cdot M_d^{-1} \in \mathbb{F}_q^d$.

For example, let $d := 4$ and $\gamma := [1, 1, 0, 0] \in \mathbb{F}_2^4$, thus $p_\gamma = X^4 + X + 1 \in \mathbb{F}_2[X]$. Since $X^2 + X + 1 \in \mathbb{F}_2[X]$ is the only irreducible polynomial of degree 2, we have the factorisations $X^3 + 1 = (X + 1)(X^2 + X + 1) \in \mathbb{F}_2[X]$ and $X^5 + 1 = (X + 1)(X^4 + X^3 + X^2 + X + 1) \in \mathbb{F}_2[X]$, and finally $X^{15} + 1 = (X+1)(X^2+X+1)(X^4+X^3+X^2+X+1) \cdot (X^4+X+1)(X^4+X^3+1) \in \mathbb{F}_2[X]$.

Thus $p_\gamma \in \mathbb{F}_2[X]$ is irreducible, and we have $p_\gamma \mid X^{15}+1$, but $p_\gamma \nmid X^k+1$ for all proper divisors k of 15.

Hence for $\kappa := [1,0,1,0] \in \mathbb{F}_2^4$ the associated keystream of period 15 is given as $[1,0,1,0,1,1,1,1,0,0,0,1,0,0,1;1,0,1,0,\ldots] \subseteq \mathbb{F}_2$. Using the initial elements of the keystream, we for $n \in \mathbb{N}$ get $\mathrm{rk}(M_n)$ as $[1,2,2,4,4,4,4,\ldots]$, and indeed $[1,1,1,1] \cdot M_4^{-1} = [1,1,0,0] = \gamma \in \mathbb{F}_2^4$, where

$$M_7 = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \quad \text{and} \quad M_4^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

7 Perfect security

(7.1) Probability spaces. a) Let $X \neq \emptyset$ be a finite set, called the **sample space**. The subsets of X are called **events**, the elements of X are called **elementary events**, the subset $\emptyset \subseteq X$ is called the **null event**, and $X \subseteq X$ is called the **certain event**. Two events $A, B \subseteq X$ such that $A \cap B = \emptyset$ are called **mutually exclusive**.

Let $\mathrm{Pot}(X)$ be the **power set** of X, i. e. the set of all events. Then a map $\mu = \mu_X \colon \mathrm{Pot}(X) \to \mathbb{R}_{\geq 0}$ is called a **probability distribution** or **measure** on X, if **i)** $\mu(X) = 1$, and **ii)** $\mu(A \cup B) = \mu(A) + \mu(B)$ whenever $A, B \subseteq X$ such that $A \cap B = \emptyset$. The set X together with the probability distribution μ is called a **probability space** or **measure space**. Given an event $A \subseteq X$, then $\mu(A) \in \mathbb{R}_{\geq 0}$ is called its **probability** or its **measure**; for $x \in X$ we also write $\mu(x) := \mu(\{x\})$.

For $A \subseteq X$ we have $\mu(X \setminus A) = \mu(X) - \mu(A) = 1 - \mu(A)$, in particular $\mu(\emptyset) = 1 - \mu(X) = 0$. Since X is finite, for $A = \coprod_{x \in A} \{x\} \subseteq X$ we have $\mu(A) = \sum_{x \in A} \mu(x)$, implying **monotonicity** $\mu(A) \leq \mu(B)$ for $A \subseteq B \subseteq X$, in particular $0 \leq \mu(A) \leq 1$. Moreover, μ is uniquely determined by the $\mu(x)$ for all elementary events $x \in X$; conversely, given $\mu(x) \in \mathbb{R}_{\geq 0}$ for all $x \in X$, this extends to a probability distribution on X if and only if $\sum_{x \in X} \mu(x) = 1$. E. g. $\mu \colon X \to \mathbb{R}_{\geq 0} \colon x \mapsto \frac{1}{|X|}$ is called the **uniform** distribution; we refer to it by making **random choices**.

b) Given probability spaces X and Y, then $X \times Y$ becomes a probability space by giving **joint probabilities** $\mu_{X \times Y}(x,y) \in \mathbb{R}_{\geq 0}$ such that $\mu_{X \times Y}(x,Y) = \sum_{y \in Y} \mu_{X \times Y}(x,y) = \mu_X(x)$ and $\mu_{X \times Y}(X,y) = \sum_{x \in X} \mu_{X \times Y}(x,y) = \mu_Y(y)$, for $x \in X$ and $y \in Y$. Hence μ_X and μ_Y can be recovered from $\mu_{X \times Y}$, and for $x \in X$ such that $\mu_X(x) = 0$, or $y \in Y$ such that $\mu_Y(y) = 0$, we have $\mu_{X \times Y}(x,y) = 0$. E. g. $X \times Y$ becomes a probability space by letting $\mu_{X \times Y}(x,y) := \mu_X(x)\mu_Y(y)$, for $x \in X$ and $y \in Y$; in this case μ_X and μ_Y are called **independent**.

The **conditional probability** for the occurrence of an event $A \subseteq X$, provided an event $B \subseteq Y$ such that $\mu_Y(B) > 0$ has already occurred, is given as $\mu_X(A|B) := \frac{\mu_{X \times Y}(A \times B)}{\mu_{X \times Y}(X \times B)} = \frac{\mu_{X \times Y}(A \times B)}{\mu_Y(B)} \in \mathbb{R}_{\geq 0}$. Then $\mu_X(\cdot|B)$, where still $\mu_Y(B) > 0$, again is a probability distribution: We have $\mu_X(A|B) = \sum_{x \in A} \mu_X(x|B)$ as well as $\sum_{x \in X} \mu_X(x|B) = \frac{1}{\mu_Y(B)} \cdot \sum_{x \in X} \mu_{X \times Y}(\{x\} \times B) = \frac{1}{\mu_Y(B)} \cdot \sum_{x \in X} \sum_{y \in B} \mu_{X \times Y}(x, y) = \frac{1}{\mu_Y(B)} \cdot \sum_{y \in B} \mu_Y(y) = 1$.

Since $\mu_X(A) = 0$ implies $\mu_{X \times Y}(A \times Y) = 0$, we in this case have $\mu_X(A|B) = 0$. If $\mu_Y(B) = 0$ then $\mu_{X \times Y}(X \times B) = 0$, hence in this case we for completeness let $\mu_X(A|B) := 0$, for $A \subseteq X$, which of course is not a probability distribution. Anyway, this yields **Bayes's Theorem**: For all $A \subseteq X$ and $B \subseteq Y$ we have $\mu_X(A|B)\mu_Y(B) = \mu_{X \times Y}(A \times B) = \mu_Y(B|A)\mu_X(A)$.

In particular, for $x \in X$ and $y \in Y$ we have $\mu_{X \times Y}(x, y) = \mu_X(x|y)\mu_Y(y)$. Thus μ_X and μ_Y are independent if and only if $\mu_X(x|y) = \mu_X(x)$ for all $x \in X$ and $y \in Y$ such that $\mu_Y(y) > 0$.

(7.2) Randomised cryptosystems. Let $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem, and let $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{K}}$ be probability distributions on $\mathcal{P}$ and $\mathcal{K}$, respectively. Typically, $\mu_{\mathcal{P}}$ is determined by properties of the plaintext language, while $\mu_{\mathcal{K}}$ is specified by the particular protocol used. To use the cryptosystem as a **randomised cryptosystem**, we assume that for each plaintext encrypted a new key is chosen, according to $\mu_{\mathcal{K}}$, where we assume that the probability distributions $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{K}}$ are independent.

We have an induced probability distribution $\mu_{\mathcal{C}}$ on $\mathcal{C}$, where for $x \in \mathcal{P}$ and $y \in \mathcal{C}$ we have $\mu_{\mathcal{C}}(y) = \mu_{\mathcal{P} \times \mathcal{K}}(\{[x, k] \in \mathcal{P} \times \mathcal{K}; E_k(x) = y\})$ and $\mu_{\mathcal{P} \times \mathcal{C}}(x, y) = \mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(\{k \in \mathcal{K}; E_k(x) = y\}|x) = \mu_{\mathcal{P} \times \mathcal{K}}(x, \{k \in \mathcal{K}; E_k(x) = y\})$. Then $\mu_{\mathcal{C}}$ is a probability distribution, and $\mu_{\mathcal{P} \times \mathcal{C}}$ is a joint probability distribution associated with $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{C}}$: We have $\mu_{\mathcal{P} \times \mathcal{C}}(x, \mathcal{C}) = \sum_{y \in \mathcal{C}} \mu_{\mathcal{P} \times \mathcal{K}}(x, \{k \in \mathcal{K}; E_k(x) = y\}) = \mu_{\mathcal{P} \times \mathcal{K}}(x, \mathcal{K}) = \mu_{\mathcal{P}}(x)$ and $\mu_{\mathcal{P} \times \mathcal{C}}(\mathcal{P}, y) = \sum_{x \in \mathcal{P}} \mu_{\mathcal{P} \times \mathcal{K}}(x, \{k \in \mathcal{K}; E_k(x) = y\}) = \mu_{\mathcal{P} \times \mathcal{K}}(\{[x, k] \in \mathcal{P} \times \mathcal{K}; E_k(x) = y\}) = \mu_{\mathcal{C}}(y)$. Since $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{K}}$ are independent we more precisely get $\mu_{\mathcal{P} \times \mathcal{C}}(x, y) = \mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(\{k \in \mathcal{K}; E_k(x) = y\})$ and $\mu_{\mathcal{C}}(y) = \sum_{x \in \mathcal{P}} \mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(\{k \in \mathcal{K}; E_k(x) = y\})$.

A ciphertext $y \in \mathcal{C}$ such that $\mu_{\mathcal{C}}(y) = 0$ does never occur, and can be discarded from $\mathcal{C}$, and similarly a plaintext $x \in \mathcal{P}$ such that $\mu_{\mathcal{P}}(x) = 0$ does never occur, and can be discarded from $\mathcal{P}$.

The cryptosystem is called **perfectly secure**, if $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{C}}$ are independent, i. e. for all $y \in \mathcal{C}$ such that $\mu_{\mathcal{C}}(y) > 0$ we have $\mu_{\mathcal{P}}(\cdot|y) = \mu_{\mathcal{P}}$, or equivalently for all $x \in \mathcal{P}$ such that $\mu_{\mathcal{P}}(x) > 0$ we have $\mu_{\mathcal{C}}(\cdot|x) = \mu_{\mathcal{C}}$. Hence from observing a ciphertext no information about the associated plaintext can be extracted.

(7.3) Theorem: Shannon [1949]. Let $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem such that $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$, together with independent probability distributions $\mu_{\mathcal{K}}$ and $\mu_{\mathcal{P}}$, such that $\mu_{\mathcal{P}}(x) > 0$ for all $x \in \mathcal{P}$. Then the following are equivalent:
a) The cryptosystem is perfectly secure, and we have $\mu_{\mathcal{C}}(y) > 0$ for all $y \in \mathcal{C}$.

b) The probability distribution $\mu_{\mathcal{K}}$ is uniform, and for all $x \in \mathcal{P}$ and $y \in \mathcal{C}$ there is a unique $k \in \mathcal{K}$ such that $E_k(x) = y$.

Proof. a)⇒b) For $x \in \mathcal{P}$ and $y \in \mathcal{C}$ we by perfect security have $\mu_{\mathcal{P}\times\mathcal{C}}(x, y) = \mu_{\mathcal{P}}(x)\mu_{\mathcal{C}}(y) > 0$, implying that there is $k \in \mathcal{K}$ such that $E_k(x) = y$. Thus fixing $x \in \mathcal{P}$ we have $\{E_k(x) \in \mathcal{C}; k \in \mathcal{K}\} = \mathcal{C}$, hence $|\mathcal{C}| = |\{E_k(x) \in \mathcal{C}; k \in \mathcal{K}\}| \leq |\mathcal{K}| = |\mathcal{C}|$, implying equality and thus $E_k(x) \neq E_{k'}(x)$ whenever $k \neq k'$, proving the second assertion.

For $x \in \mathcal{P}$ and $y \in \mathcal{C}$ let $k_{x,y} \in \mathcal{K}$ be the unique key such that $E_{k_{x,y}}(x) = y$. Hence $\mu_{\mathcal{P}}(x)\mu_{\mathcal{C}}(y) = \mu_{\mathcal{P}\times\mathcal{C}}(x, y) = \mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(k_{x,y})$ implies $\mu_{\mathcal{K}}(k_{x,y}) = \mu_{\mathcal{C}}(y)$. By the injectivity of encryption functions we have $k_{x,y} \neq k_{x',y}$ whenever $x \neq x'$, hence by $|\mathcal{P}| = |\mathcal{K}|$ we have $\{k_{x,y} \in \mathcal{K}; x \in \mathcal{P}\} = \mathcal{K}$. Thus for all $k \in \mathcal{K}$ and $y \in \mathcal{C}$ we have $\mu_{\mathcal{K}}(k) = \mu_{\mathcal{C}}(y)$, proving the first assertion.

b)⇒a) Keeping the above notation we have $\mu_{\mathcal{K}}(k_{x,y}) = \frac{1}{|\mathcal{K}|}$ for all $x \in \mathcal{P}$ and $y \in \mathcal{C}$, and thus $\mu_{\mathcal{C}}(y) = \sum_{x\in\mathcal{P}} \mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(k_{x,y}) = \frac{1}{|\mathcal{K}|} \cdot \sum_{x\in\mathcal{P}} \mu_{\mathcal{P}}(x) = \frac{1}{|\mathcal{K}|} > 0$. This yields $\mu_{\mathcal{P}}(x|y) = \frac{\mu_{\mathcal{P}}(x)\mu_{\mathcal{K}}(k_{x,y})}{\mu_{\mathcal{C}}(y)} = \mu_{\mathcal{P}}(x)$ for all $x \in \mathcal{P}$ and $y \in \mathcal{C}$. ♯

(7.4) Corollary. a) Condition a) implies that $\mu_{\mathcal{C}}$ is uniform.
b) The validity of condition a) does not depend on the particular choice of $\mu_{\mathcal{P}}$, as long as $\mu_{\mathcal{P}}(x) > 0$ for all $x \in \mathcal{P}$.
c) If we allow $|\mathcal{P}|$, $|\mathcal{C}|$, $|\mathcal{K}|$ to be distinct then condition a) still implies $|\mathcal{C}| \leq |\mathcal{K}|$; by the injectivity of encryption functions we have $|\mathcal{P}| \leq |\mathcal{C}|$ anyway.

(7.5) Vernam's One-Time Pad [1917]. We consider the Vigenère cipher over $R := \mathbb{Z}_n$, where $n \in \mathbb{N}$, of block length $l \in \mathbb{N}$; the particular case $n = 2$ being **Vernam's One-Time Pad**: We have $\mathcal{P} = \mathcal{C} = \mathcal{K} = R^l$, and for $k \in R^l$ we have $E_k\colon v \mapsto v + k$ and $D_k\colon v \mapsto v - k$. Since $k = E_k(v) - v \in R^l$, for $v \in R^l$, the Vigenère cipher is vulnerable to a known-plaintext attack. Thus a chosen key may only be used to encrypt a single plaintext, and has to be chosen anew afterwards; the terminology 'one-time' is reminiscent of this fact.

We have $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}| = R^l$, and for $v, w \in R^l$ the unique key such that $E_k(v) = w$ is given as $k := w - v \in R^l$. Hence if $\mu_{\mathcal{K}}$ is uniform, and if $\mu_{\mathcal{P}}$ is any probability distribution such that $\mu_{\mathcal{P}}(x) > 0$ for all $x \in \mathcal{P}$, then the Vigenère cipher is perfectly secure.

(7.6) Remark. For further aspects of information theory related to cryptography, in particular the notion of **entropy** [Shannon, 1948], see [12, Ch.2.4ff.].

II Public key cryptography

8 The RSA cryptosystem

(8.1) Cyclic groups. Let G be a finite group and $U \leq G$. Then for the associated **group orders** we have **Lagrange's Theorem** $|U| \mid |G|$; the number $[G \colon U] := \frac{|G|}{|U|} \in \mathbb{N}$ is called the **index** of U in G:

For $g \in G$ let $Ug := \{ug \in G; h \in U\} \subseteq G$ be the associated **(right) coset**. We have $|Ug| = |U|$: For $u, v \in U$, from $ug = vg$ we get $u = ugg^{-1} = vgg^{-1} = v$. We have $G = \bigcup_{g \in G} Ug$, and given $g, h \in G$ from $Ug \cap Uh \neq \emptyset$ we already get $Ug = Uh$: Let $vh \in Ug \cap Uh$ for some $v \in U$, then for all $u \in U$ we have $uh = uv^{-1}vh \in Ug$, thus $Uh \subseteq Ug$, and similarly $Ug \subseteq Uh$. Thus we have $G = \coprod_{t \in \mathcal{T}} Ut$ for a suitable **(right) transversal** $\mathcal{T} \subseteq G$. $\sharp$

Given $g \in G$ then $\langle g \rangle := \{g^k \in G; k \in \mathbb{Z}\} \leq G$ is the smallest subgroup of G containing g. The number $|g| := |\langle g \rangle| \in \mathbb{N}$ is called the **order** of g; in particular we have $|g| \mid |G|$. The group G is called **cyclic**, if there is $g \in G$ such that $G = \langle g \rangle$; if $|G| = n \in \mathbb{N}$ we write $G \cong C_n$. E. g. we have $(\mathbb{Z}, +) = \langle 1 \rangle = \langle -1 \rangle$, and for $n \in \mathbb{Z}$ we have $(\mathbb{Z}/n\mathbb{Z}, +) = \langle 1 \rangle$.

(8.2) Theorem. Let G be a finite group.
a) Let $g \in G$ and $I_g := \{i \in \mathbb{Z}; g^i = 1\}$. Then we have $I_g = |g|\mathbb{Z}$ and $\langle g \rangle = \{g^i \in G; i \in \{0, \ldots, |g|-1\}\}$; in particular we have $g^{|g|} = g^{|G|} = 1$.
b) Let G be cyclic. Then any subgroup of G is cyclic as well. There is $U \leq G$ of order $d \in \mathbb{N}$ if and only if $d \mid |G|$; in this case U is uniquely determined.
c) G is cyclic if and only if G has at most one subgroup of order d for any $d \in \mathbb{N}$. In particular, if $|G|$ is a prime then G is cyclic.

Proof. a) There are $i \neq j$ such that $g^i = g^j$, hence $g^{i-j} = 1$, thus $0 \neq i-j \in I_g$. Let $n \in \mathbb{N}$ be minimal such that $n \in I_g$, then we have $I_g = n\mathbb{Z}$: We have $n\mathbb{Z} \subseteq I_g$, and by quotient and remainder for $i = nq + r \in I_g$, where $q, r \in \mathbb{Z}$ such that $r \in \{0, \ldots, n-1\}$, we have $g^r = g^i(g^n)^{-q} = 1$, hence by the choice of n we deduce $r = 0$ and thus $i \in n\mathbb{Z}$. Moreover, for any $i = nq + r \in \mathbb{Z}$ we have $g^r = g^i$, and since $g^i = g^j$ implies $i - j \in I_g = n\mathbb{Z}$, we get $\langle g \rangle = \{g^i \in G; i \in \{0, \ldots, |g|-1\}\}$.

b) Let $G = \langle g \rangle$. For $\{1\} \neq U \leq G$ let $I_U := \{k \in \mathbb{Z}; g^k \in U\}$. Then letting $m \in \mathbb{N}$ be minimal such that $m \in I_U$, we have $\langle g^m \rangle \leq U$. Conversely, if $g^i \in U$ for some $i = mq + r \in \mathbb{Z}$, where $q, r \in \mathbb{Z}$ such that $r \in \{0, \ldots, m-1\}$, we get $g^r = g^i(g^m)^{-q} \in U$, thus $r \in I_U$ and hence by the choice of m we deduce $r = 0$ and thus $g^i \in \langle g^m \rangle$. Hence $U = \langle g^m \rangle$ is cyclic.

Let $n := |G| = |g|$ and let $k \in \mathbb{Z}$. Then for $i \in \mathbb{Z}$ we have $g^{ik} = 1$ if and only if $n \mid ik$, which holds if and only if $\frac{n}{\gcd(k,n)} \mid i$. Hence we have $|g^k| = \frac{n}{\gcd(k,n)}$; in particular we have $|g^k| = n$ if and only if $\gcd(k, n) = 1$. Hence for any $d \in \mathbb{N}$ such that $d \mid n$ there is a subgroup of order d: Writing $n = md$ we have $|g^m| = d$. Finally, if $|g^k| = d$ for $k \in \mathbb{Z}$, then $\frac{n}{\gcd(k,n)} = d = \frac{n}{m}$ implies $m = \gcd(k, n) \mid k$, thus $\langle g^k \rangle \leq \langle g^m \rangle$, implying equality.

c) If H is a cyclic group of order $n \in \mathbb{N}$, then using Euler's totient function there are precisely $\varphi(n) \in \mathbb{N}$ elements $h \in H$ such that $\langle h \rangle = H$. Thus the subgroup structure of cyclic groups implies $\sum_{d \in \mathbb{N}, d \mid n} \varphi(d) = n$ for any $n \in \mathbb{N}$.

Let G fulfil the assumption on the subgroup structure, and let $n := |G|$. For any $d \in \mathbb{N}$ there is an element of order d only if $d \mid n$, and if there is an element of order d there are precisely $\varphi(d)$ of them. Thus by $\sum_{d \mid n, d \neq n} \varphi(d) = n - \varphi(n) < n$ there is an element of order n. ♯

(8.3) Corollary. a) Let $\mathrm{Aut}(G)$ be the **automorphism group** of G, i. e. the set of all bijective group homomorphisms $G \to G$ together with the composition of maps. If $G = \langle g \rangle$ is a cyclic group of order $n \in \mathbb{N}$, then any $\varphi \in \mathrm{Aut}(G)$ is uniquely determined by $\varphi(g) \in G$, which again is a generator, implying that there is $k \in \mathbb{Z}_n^*$ such that $\varphi(g) = g^k$; since $g^k = g^{k+in}$, for all $k, i \in \mathbb{Z}$, we may assume $k \in (\mathbb{Z}/n\mathbb{Z})^*$. Conversely, if $k \in (\mathbb{Z}/n\mathbb{Z})^*$ then g^k is a generator of G, and there is $\varphi_k \in \mathrm{Aut}(\langle g \rangle)$ such that $\varphi_k(g) = g^k$. Thus we have a group isomorphism $(\mathbb{Z}/n\mathbb{Z})^* \to \mathrm{Aut}(G) \colon k \mapsto \varphi_k$.

b) Let $n \in \mathbb{N}$. From $|(\mathbb{Z}/n\mathbb{Z})^*| = \varphi(n)$ we conclude **Euler's Theorem**: For all $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $x^{\varphi(n)} = 1$. If $p \in \mathbb{N}$ is a prime, then in particular we have **Fermat's Theorem**: For all $x \in \mathbb{Z}/p\mathbb{Z}$ we have $x^p = x$: If $x \in (\mathbb{Z}/p\mathbb{Z})^* = (\mathbb{Z}/p\mathbb{Z}) \setminus \{0\}$, then we get $x^{p-1} = 1$, and if $x = 0$ then $x^p = 0 = x$ anyway.

c) Let $\mathbb{F}_q$ be the field with q elements. Then we have **Artin's Theorem**: The group $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ is cyclic; an element $\rho \in \mathbb{F}_q^*$ such that $\langle \rho \rangle = \mathbb{F}_q^*$ is called a **primitive root**: For all $d \in \mathbb{N}$ such that $d \mid q - 1 = |\mathbb{F}_q^*|$ the elements of $\mathbb{F}_q^*$ having order dividing d are roots of $X^d - 1 \in \mathbb{F}_q[X]$. Since there are at most d roots in the field $\mathbb{F}_q$, we conclude that $\mathbb{F}_q^*$ has at most d elements having order dividing d, thus has at most one subgroup of order d, hence is cyclic.

(8.4) The Rivest-Shamir-Adleman (RSA) cryptosystem [1978]. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$ be the associated **modulus**. Let $\mathcal{P} = \mathcal{C} := (\mathbb{Z}/n\mathbb{Z})^*$ and $\mathcal{K} := (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, and let the encryption functions $\mathcal{E} = \{E_e \colon \mathcal{P} \to \mathcal{C}; e \in \mathcal{K}\}$ and decryption functions $\mathcal{D} = \{D_d \colon \mathcal{C} \to \mathcal{P}; d \in \mathcal{K}\}$ be given as follows: For $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ let $E_e \colon (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^* \colon a \mapsto a^e$, and similarly for $d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ let $D_d \colon (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^* \colon a \mapsto a^d$. Given $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, choosing $d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ such that $E_e D_d = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$, the **public key** is $[n, e]$, the **private key** is $[p, q, d]$. We show that this is an unsymmetric cryptosystem:

Indeed, for $e \in \mathbb{Z}/\varphi(n)\mathbb{Z}$ the function E_e is well-defined: Since $|(\mathbb{Z}/n\mathbb{Z})^*| = \varphi(n)$, by Euler's Theorem we have $a^{i+j\varphi(n)} = a^i$, for $a \in (\mathbb{Z}/n\mathbb{Z})^*$ and $i, j \in \mathbb{Z}$. Next, for $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ the function E_e is bijective, the inverse being D_d for $d := e^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$: We have $ed = 1 \in \mathbb{Z}/\varphi(n)\mathbb{Z}$, and thus for $a \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $D_d(E_e(a)) = (a^e)^d = a^{ed} = a^1 = a \in (\mathbb{Z}/n\mathbb{Z})^*$. Hence this is a cryptosystem. Since given an encryption key $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ the decryption key $d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ is computed by an inversion modulo $\varphi(n)$, which is kept private, this is an unsymmetric cryptosystem.: $\sharp$

For further analysis, in particular to determine $\varphi(n)$, recall that by the Chinese remainder theorem the natural map $\nu_{p,q} \colon \mathbb{Z}/n\mathbb{Z} \to (\mathbb{Z}/p\mathbb{Z}) \oplus (\mathbb{Z}/q\mathbb{Z}) \colon a + n\mathbb{Z} \mapsto [a + p\mathbb{Z}, a + q\mathbb{Z}]$ is a ring isomorphism, where $(\mathbb{Z}/p\mathbb{Z}) \oplus (\mathbb{Z}/q\mathbb{Z})$ is the Cartesian product $(\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ becoming a commutative ring with respect to componentwise addition and multiplication. In particular, $\nu_{p,q}$ induces a group isomorphism $(\mathbb{Z}/n\mathbb{Z})^* \cong (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z})^*$, where the right hand side becomes a group with respect to componentwise multiplication. Hence we get $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$. We derive a couple of further consequences:

For $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z}) \setminus (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ the function E_e actually is not bijective: From $\gcd(e, \varphi(n)) > 1$ we may assume that $f := \gcd(e, p-1) > 1$, hence letting $a \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root we have $a^{\frac{p-1}{f}} \neq 1$ and $(a^{\frac{p-1}{f}})^e = (a^{p-1})^{\frac{e}{f}} = 1 = 1^e$.

Moreover for $e \in \mathbb{Z}/\varphi(n)\mathbb{Z}$ the map $E_e \colon (\mathbb{Z}/n\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^*$ can be naturally extended to map $E_e \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \colon a \mapsto a^e$, as soon as we restrict ourselves to choose residue class representatives $e \in \mathbb{N}$, thus defining a cryptosystem on $\mathbb{Z}/n\mathbb{Z}$ (although this will turn out to be useless): Using the natural map $\nu_{p,q}$, the elements of $(\mathbb{Z}/n\mathbb{Z}) \setminus (\mathbb{Z}/n\mathbb{Z})^*$ are given as $[a, 0]$ for $a \in (\mathbb{Z}/p\mathbb{Z})^*$, or $[0, a]$ for $a \in (\mathbb{Z}/q\mathbb{Z})^*$, or $[0, 0]$; since $\varphi(p) \mid \varphi(n)$ and $\varphi(q) \mid \varphi(n)$, arguing as above shows that E_e is well-defined and bijective, and that for the analogously extended maps D_d we have $E_e D_d = \mathrm{id}_{\mathbb{Z}/n\mathbb{Z}}$ if and only if $E_e D_d|_{(\mathbb{Z}/n\mathbb{Z})^*} = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$

(8.5) Encryption functions of the RSA cryptosystem. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$.

a) We determine $|\mathcal{E}| = |\mathcal{D}|$, that is how many distinct encryption and decryption function there actually are:

Since for $e, e' \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ we have $E_e E_{e'} = E_{ee'} \colon a \mapsto (a^e)^{e'} = a^{ee'}$, we have a group homomorphism $\epsilon \colon (\mathbb{Z}/\varphi(n)\mathbb{Z})^* \to \mathrm{Aut}((\mathbb{Z}/n\mathbb{Z})^*) \colon e \mapsto E_e$. Using the natural embeddings $(\mathbb{Z}/p\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^*$ and $(\mathbb{Z}/q\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^*$ shows that E_e leaves $(\mathbb{Z}/p\mathbb{Z})^*$ and $(\mathbb{Z}/q\mathbb{Z})^*$ invariant, hence by restriction induces automorphisms thereon, and moreover is uniquely determined by these restrictions. Thus we actually have $\epsilon \colon (\mathbb{Z}/\varphi(n)\mathbb{Z})^* \to \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \times \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \leq \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z})^*) \cong \mathrm{Aut}((\mathbb{Z}/n\mathbb{Z})^*)$.

Since we have $\mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \cong \mathrm{Aut}(C_{p-1}) \cong (\mathbb{Z}/(p-1)\mathbb{Z})^*$, and similarly $\mathrm{Aut}((\mathbb{Z}/q\mathbb{Z})^*) \cong \mathrm{Aut}(C_{q-1}) \cong (\mathbb{Z}/(q-1)\mathbb{Z})^*$, we have $E_e = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$ if and only if $e \equiv 1 \pmod{p-1}$ and $e \equiv 1 \pmod{q-1}$, which holds if and only if $e \equiv 1 \pmod{\mathrm{lcm}(p-1, q-1)}$; note that this condition is independent of the choice of representatives modulo $\varphi(n)$. Hence we have $\ker(\epsilon) := \{e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*; E_e = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}\} = \{e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*; \mathrm{lcm}(p-1, q-1) \mid e - 1\}$. By the homomorphism theorem (which we do not prove here) this can be rephrased as $\mathcal{E} = \mathrm{im}(\epsilon) \cong (\mathbb{Z}/\varphi(n)\mathbb{Z})^* / \ker(\epsilon) \cong (\mathbb{Z}/\mathrm{lcm}(p-1, q-1)\mathbb{Z})^*$, showing that

$|\mathcal{E}| = \varphi(\mathrm{lcm}(p-1,q-1))$. Alternatively, we may argue as follows:

Since the divisibility condition $\mathrm{lcm}(p-1,q-1) \mid e-1$ implies $\gcd(e,\varphi(n)) = 1$, we have $\ker(\epsilon) = \{e \in \mathbb{Z}/\varphi(n)\mathbb{Z}; \mathrm{lcm}(p-1,q-1) \mid e-1\}$. Next, since $\mathrm{lcm}(p-1,q-1) \mid \varphi(n)$ we thus have $|\ker(\epsilon)| = \frac{\varphi(n)}{\mathrm{lcm}(p-1,q-1)} = \frac{(p-1)(q-1)}{\mathrm{lcm}(p-1,q-1)} = \gcd(p-1,q-1)$. Since we have $E_e = E_{e'}$ if and only if $E_{e^{-1}e'} = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$, which holds if and only if $e^{-1}e' \in \ker(\epsilon) \leq (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, that is $\ker(\epsilon)e = \ker(\epsilon)e' \subseteq (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, by Lagrange's Theorem we infer $|\mathcal{E}| = |\mathrm{im}(\epsilon)| = \frac{|(\mathbb{Z}/\varphi(n)\mathbb{Z})^*|}{|\ker(\epsilon)|} = \frac{\varphi((p-1)(q-1))}{\gcd(p-1,q-1)}$. Finally, since for any prime $r \in \mathbb{N}$ and $i > j \in \mathbb{N}_0$ we have $\frac{\varphi(r^i r^j)}{\gcd(r^i,r^j)} = \frac{r^{i+j-1}(r-1)}{r^j} = r^{i-1}(r-1) = \varphi(r^i) = \varphi(\mathrm{lcm}(r^i,r^j))$, we in conclusion get $|\mathcal{E}| = \varphi(\mathrm{lcm}(p-1,q-1))$. ♯

Hence p and q have to be chosen such that $\gcd(p-1,q-1) = \frac{(p-1)(q-1)}{\mathrm{lcm}(p-1,q-1)}$ is not too large, otherwise the set $\mathcal{E}$ of actually available encryption functions might become too small. Thus for a good choice of p and q we have $\gcd(p-1,q-1) = 2$, and moreover the numbers $p-1$ and $q-1$ do not have too many small prime divisors, so that we expect that $|\mathcal{E}| = \varphi(\mathrm{lcm}(p-1,q-1)) = \varphi(\frac{1}{2}(p-1)(q-1))$ is a number of magnitudse $\sim pq = n$

For an encryption key $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ an element $d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ is a decryption key if and only if $E_e D_d = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$, that is $ed \in \ker(\epsilon)$, or equivalently $d \in \ker(\epsilon)e^{-1}$. Hence there are precisely $|\ker(\epsilon)| = \gcd(p-1,q-1)$ possible decryption keys, since $\ker(\epsilon) = \{c \in \mathbb{Z}/\varphi(n)\mathbb{Z}; c \equiv 1 \pmod{\mathrm{lcm}(p-1,q-1)}\}$ given as $\{e^{-1}\cdot\big(1+k\cdot\mathrm{lcm}(p-1,q-1)\big) \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*; k \in \{0,\ldots,\gcd(p-1,q-1)-1\}\}$. In particular, the above condition on good choices of p and q also entails that the number of possible decryption keys for a given encryption key is as small as possible. Indeed, since the more possible decryption keys there are, the easier it becomes to break the cryptosystem, this number should be kept small.

b) In the same vein, we compare the order $|e| \in \mathbb{N}$ of $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ and the order $k := |E_e| \in \mathbb{N}$ of $E_e \in \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \times \mathrm{Aut}((\mathbb{Z}/q\mathbb{Z})^*) \leq \mathrm{Aut}((\mathbb{Z}/n\mathbb{Z})^*)$. Using the group homomorphism $\epsilon\colon (\mathbb{Z}/\varphi(n)\mathbb{Z})^* \to \mathrm{Aut}((\mathbb{Z}/n\mathbb{Z})^*)\colon e \mapsto E_e$, we get $k \mid |e|$, and since $e^k \in \ker(\epsilon)$ Euler's Theorem implies $|e| \mid k\cdot|\ker(\epsilon)|$. Thus if $|\ker(\epsilon)| = \gcd(p-1,q-1)$ is chosen to be small, then k is of magnitude $\sim |e|$.

Now k is the order of $e \in (\mathbb{Z}/\mathrm{lcm}(p-1,q-1)\mathbb{Z})^*$, but we may determine k also as follows: Let $l,m \in \mathbb{N}$ be the order of $e \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ and $e \in (\mathbb{Z}/(q-1)\mathbb{Z})^*$, respectively. Hence these are are the order of $E_e \in \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*)$ and $E_e \in \mathrm{Aut}((\mathbb{Z}/q\mathbb{Z})^*)$, respectively, and we have $k = \mathrm{lcm}(l,m)$.

This reduces the question of finding k to finding the order l of $e \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$. Instead of successively computing e^i, for $i \geq 1$ increasing until we find $e^l = 1$, we proceed as follows, recalling that computing powers with respect to a known exponent can be done by repeated squaring: If the factorisation of $p-1$ is known, then $\varphi(p-1)$ can be determined, and if the factorisation of $\varphi(p-1)$ is known as well, using $l \mid \varphi(p-1)$ we infer that only the divisors of $\varphi(p-1)$ are candidates for l to be checked.

(8.6) Example. Let $p := 97$ and $q := 193$, hence $n := pq = 18721$. Letting $e := 43$, using $\varphi(n) = (p-1)(q-1) = 96 \cdot 192 = 18432$ the extended Euclidean algorithm yields $1 = \gcd(e, \varphi(n)) = -8573 \cdot e + 20 \cdot \varphi(n)$, hence $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ and $d := e^{-1} = -8573 = 9859 \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$.

i) We determine the number of encryption functions, where it turns out that this example is chosen as badly as possible: We have $p-1 = 2^5 \cdot 3$ and $q-1 = 2^6 \cdot 3$, thus $p-1 \mid q-1$, hence $\gcd(p-1, q-1) = p-1 = 96$ and $\operatorname{lcm}(p-1, q-1) = q-1 = 192$. Hence $(\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ has order $\varphi(\varphi(n)) = \varphi((p-1)(q-1)) = \varphi(2^{11} \cdot 3^2) = 2^{11} \cdot 3 = 6144$, but there are only $|\mathcal{E}| = \varphi(\operatorname{lcm}(p-1, q-1)) = \varphi(2^6 \cdot 3) = 2^6 = 64$ encryption functions, which is far off the number we had hoped for.

Moreover, given $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, there are $|\ker \epsilon| = \gcd(p-1, q-1) = 96$ decryption keys in $(\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, which since $\ker(\epsilon) = \{c \in \mathbb{Z}/\varphi(n)\mathbb{Z}; c \equiv 1 \pmod{\operatorname{lcm}(p-1, q-1)}\} = \{c \in \mathbb{Z}/\varphi(n)\mathbb{Z}; c \equiv 1 \pmod{192}\}$ are given as $\{e^{-1} \cdot (1 + k \cdot 192) \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*; k \in \{0, \ldots, 95\}\}$. For example, for $e := 43$ and $d := 9859$ we get $\{9859, 9859 \cdot (1 + 192), \ldots, 9859 \cdot (1 + 95 \cdot 192)\} = \{67, 259, \ldots, 18307\} \subseteq (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, hence we might also choose $d' := 67 \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ as a decryption key.

ii) Using the structure of groups of prime residues (which is not proved here) we get $(\mathbb{Z}/\varphi(n)\mathbb{Z})^* \cong (\mathbb{Z}/(p-1)(q-1)\mathbb{Z})^* \cong (\mathbb{Z}/2^{11}\mathbb{Z})^* \times (\mathbb{Z}/3^2\mathbb{Z})^* \cong (C_2 \times C_{2^9}) \times (C_2 \times C_3)$. Hence any element $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ has order dividing $2^9 \cdot 3 = 1536$, where the latter maximum is attained by a fraction of $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3} = \frac{2048}{6144}$ of all elements; for example, by $e = 43$.

In contrast, the order of $E_e \in \operatorname{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \times \operatorname{Aut}((\mathbb{Z}/q\mathbb{Z})^*) \leq \operatorname{Aut}((\mathbb{Z}/n\mathbb{Z})^*)$ equals the order of $e \in (\mathbb{Z}/\operatorname{lcm}(p-1, q-1)\mathbb{Z})^* = (\mathbb{Z}/(q-1)\mathbb{Z})^* \cong (\mathbb{Z}/2^6\mathbb{Z})^* \times (\mathbb{Z}/3\mathbb{Z})^* \cong (C_2 \times C_{2^4}) \times C_2$. Alternatively, it is given as $|E_e| = \operatorname{lcm}(l, m)$, where in turn $l, m \in \mathbb{N}$ are the order of $e \in (\mathbb{Z}/(p-1)\mathbb{Z})^* \cong (\mathbb{Z}/2^5\mathbb{Z})^* \times (\mathbb{Z}/3\mathbb{Z})^* \cong (C_2 \times C_{2^3}) \times C_2$ and $e \in (\mathbb{Z}/(q-1)\mathbb{Z})^* \cong (\mathbb{Z}/2^6\mathbb{Z})^* \times (\mathbb{Z}/3\mathbb{Z})^* \cong (C_2 \times C_{2^4}) \times C_2$, respectively. Since $p-1 \mid q-1$ we have $l \mid m$, hence $|E_e| = m$, which coincides with the order of $e \in (\mathbb{Z}/\operatorname{lcm}(p-1, q-1)\mathbb{Z})^*$. Thus any $E_e \in \operatorname{Aut}((\mathbb{Z}/n\mathbb{Z})^*)$ has order dividing $2^4 = 16$, where the latter maximum is attained by a fraction of $\frac{1}{2} = \frac{32}{64}$ of all elements; for example, by $e = 43$.

(8.7) RSA block ciphers. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$. We obtain a block cipher of block length $l := \lfloor \log_{26}(n) \rfloor \in \mathbb{N}$ as follows: Words in $\mathcal{X}_{\text{latin}}^l$ are first encoded into $\mathbb{Z}_{26}^l$, and then via 26-adic expansion considered as elements of $\mathbb{Z}_{26^l} \subseteq \mathbb{Z}_n$; for example, for $l = 3$ the word `abc` $\in \mathcal{X}_{\text{latin}}^3$ yields $0 \cdot 26^2 + 1 \cdot 26 + 2 = 28 \in \mathbb{Z}_n$. Thus the set of admissible plaintexts can be identified with $\mathcal{P} = \mathbb{Z}_{26^l}$.

But note that using this simple idea we might have $\mathcal{P} \cap (\mathbb{Z}_n \setminus \mathbb{Z}_n^*) \neq \emptyset$. Hence the latter cases have to be excluded explicitly; these would indeed break the cryptosystem, see (9.2), but in practice they are easily avoided. Moreover, $\mathcal{P}$ must be chosen large enough, since otherwise a protocol failure results from encrypting all plaintexts using the public key, allowing to just read off the

Table 8: RSA block cipher.

$\mathcal{X}^3_{\text{latin}}$	$\mathbb{Z}_{26}$			$\mathcal{P} = \mathbb{Z}_{26^3}$	$\mathcal{C} = \mathbb{Z}^*_{18721}$
she	18	7	4	12354	13130
has	7	0	18	4750	95
sen	18	4	13	12285	7342
sed	18	4	3	12275	13805
ach	0	2	7	59	8347
ang	0	13	6	344	10022
ein	4	8	13	2925	5164
the	19	7	4	13030	13434
wea	22	4	0	14976	18716
the	19	7	4	13030	13434
rzz	17	25	25	12167	14498

plaintext associated with a given ciphertext; for example, such a protocol failure results from letting the block length to be $l := 1$ instead.

For the above example, $p := 97$ and $q := 193$, we have $n := pq = 18721$, and since $26^3 = 17576$ we let $l := 3$. Letting $e := 43$ again, the plaintext `she has sensed a change in the weather`, padded to obtain a plaintext of length divisible by l, yields the ciphertext as shown in Table 8, where we are indeed lucky enough to end up in $(\mathbb{Z}/n\mathbb{Z})^*$.

9 Attacking the RSA cryptosystem

(9.1) Protocol failure of the RSA cryptosystem. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$.

a) Multiplicativity attack. For $e \in \mathbb{Z}/\varphi(n)\mathbb{Z}$ and $x, y \in \mathbb{Z}/n\mathbb{Z}$ we have $(xy)^e = x^e y^e$. This leads to the following adaptive chosen-ciphertext attack, where an opponent, Oscar say, is able to convince Alice, who keeps the secret key, to decrypt any ciphertext Oscar's choice, except x; for example, in an identification protocol: Let $e, d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ such that $E_e D_d = \mathrm{id}_{(\mathbb{Z}/n\mathbb{Z})^*}$, and let $x \in (\mathbb{Z}/n\mathbb{Z})^*$ be a ciphertext Oscar wants to decrypt. Now Oscar chooses $y \in (\mathbb{Z}/n\mathbb{Z})^*$, and masks x by letting $z := xy^e \in (\mathbb{Z}/n\mathbb{Z})^*$, then Alice is happy to compute $z^d \in (\mathbb{Z}/n\mathbb{Z})^*$, and finally Oscar unmasks by computing $z^d y^{-1} = x^d y^{ed-1} = x^d \in (\mathbb{Z}/n\mathbb{Z})^*$, which is the decryption of x.

To prevent a multiplicativity attack, plaintexts are chosen from a suitable (large enough) admissible subset such that it is unlikely that the product of two admissible plaintexts again is admissible; for example, it may be required that, in binary representation, the first and last bits of a plaintext are identical.

b) Cycling attack. For $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ let $k := |E_e| \in \mathbb{N}$ be the order of

$E_e \in \mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \times \mathrm{Aut}((\mathbb{Z}/q\mathbb{Z})^*) \leq \mathrm{Aut}((\mathbb{Z}/n\mathbb{Z})^*)$. Since $(x^e)^{e^{k-1}} = x^{e^k} = x$, for all $x \in (\mathbb{Z}/n\mathbb{Z})^*$, if k is small the plaintext x can be recovered from the associated ciphertext x^e by repeated re-encryption. Thus e has to be chosen such that k is sufficiently large. In practice this is easily achieved:

Recall that if $l, m \in \mathbb{N}$ are the order of $e \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ and $e \in (\mathbb{Z}/(q-1)\mathbb{Z})^*$, respectively, then we have $k = \mathrm{lcm}(l, m)$. Hence we have to ensure that the latter orders are large enough. Now, if $p' \mid p-1$ is a (large) prime divisor, then $\mathbb{Z}/(p-1)\mathbb{Z}$ has a direct summand $\mathbb{Z}/p'\mathbb{Z}$, hence $\mathrm{Aut}((\mathbb{Z}/p\mathbb{Z})^*) \cong (\mathbb{Z}/(p-1)\mathbb{Z})^*$ has a direct factor $(\mathbb{Z}/p'\mathbb{Z})^* \cong C_{p'-1}$, If moreover $p'' \mid p'-1$ is a (large) prime divisor, then $C_{p'-1}$ has a unique quotient $C_{p''}$. Hence choosing $e \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ randomly, the probability that $p'' \mid l$ is given as $\frac{p''-1}{p''} = 1 - \frac{1}{p''}$. Thus if p'' is large enough, then it is (extremely) likely that l is large enough as well.

c) Common modulus attack. Let $e_1, e_2 \in \mathbb{N}$ such that $\gcd(e_1, e_2) = 1$, and let $f_1, f_2 \in \mathbb{Z}$ be Bézout coefficients such that $e_1 f_1 + e_2 f_2 = 1$. Thus we have $x = x^{e_1 f_1 + e_2 f_2} = (x^{e_1})^{f_1} \cdot (x^{e_2})^{f_2} \in (\mathbb{Z}/n\mathbb{Z})^*$. Hence, for suitable encryption keys e_1 and e_2, a plaintext x can be recovered from two associated ciphertexts x^{e_1} and x^{e_2}. To prevent a common-modulus attack, no two plaintexts encrypted must be equal; for example, by chossing parts of plaintexts randomly.

d) Low exponent attack. Let $e \in \mathbb{N}$, and let $n_1, \ldots, n_e \in \mathbb{N}$ be pairwise coprime moduli. Let $x \in \mathbb{N}$ such that $x < \min\{n_1, \ldots, n_e\}$, and let $y_i \in (\mathbb{Z}/n_i\mathbb{Z})^*$ such that $x^e = y_i \in (\mathbb{Z}/n_i\mathbb{Z})^*$ for $i \in \{1, \ldots, e\}$; hence the y_i are various ciphertexts obtained from the same plaintext x with the same encryption key e. Let $m := \prod_{i=1}^e n_i \in \mathbb{N}$, and using the Chinese remainder theorem yielding the ring isomorphism $\mathbb{Z}/m\mathbb{Z} \cong \prod_{i=1}^e \mathbb{Z}/n_i\mathbb{Z}$, let $x' \in \mathbb{Z}_m$ be the unique element such that $x' = y_i \in \mathbb{Z}/n_i\mathbb{Z}$, for $i \in \{1, \ldots, e\}$. Since we also have $x^e \in \mathbb{Z}_m$, we conclude $x' = x^e$, hence the given plaintext can be recovered as $x = \sqrt[e]{x'} \in \mathbb{N}$.

Thus e has to be chosen large enough, or again no two plaintexts encrypted must be equal. Since the number of arithmetical operations to compute $x^e \in (\mathbb{Z}/n\mathbb{Z})^*$ from $x \in (\mathbb{Z}/n\mathbb{Z})^*$ should be kept small, typical choices are $e := 3$ (for randomised plaintexts) or $e := 2^{16} + 1 = 65537$ (recalling that computing powers can be done by repeated squaring), provided these are coprime to $\varphi(n)$.

(9.2) Breaking the RSA cryptosystem. a) Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$. If $\varphi(n)$ is known, then inverses in $(\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ can be computed in polynomial time, by the extended Euclidean algorithm. a Computing $\varphi(n) = (p-1)(q-1)$ is polynomial time equivalent to factoring $n = pq$: If p and q are known, then $\varphi(n) = (p-1)(q-1)$ can be computed as well. Conversely, since factoring $n = pq$ is polynomial time equivalent to computing a prime divisor of n, if $\varphi(n)$ is known then $\varphi(n) = (p-1)(q-1) = (p-1)(\frac{n}{p} - 1)$ yields $p^2 + (\varphi(n) - n - 1)p + n = 0$, and thus $\{p, q\} = \{\frac{n+1-\varphi(n)}{2} \pm \frac{\sqrt{(n+1-\varphi(n))^2 - 4n}}{2}\}$, where $\sqrt{(n+1-\varphi(n))^2 - 4n} \in \mathbb{N}$.

In particular, for $0 \neq x \in (\mathbb{Z}/n\mathbb{Z}) \setminus (\mathbb{Z}/n\mathbb{Z})^*$ we have $1 < \gcd(x, n) < n$, thus we have found a prime divisor of n. Thus these elements are not admissible plaintexts, but since $\frac{n-1-\varphi(n)}{n} = \frac{p+q}{pq} = \frac{1}{p} + \frac{1}{q}$ they are rare anyway.

Hence breaking the RSA cryptosystem polynomial time reduces to factoring n, thus the RSA cryptosystem is secure only if factoring $n = pq$ is computationally difficult. It is conjectured that factoring integers of the form pq is as difficult as factoring arbitrary integers, and that integer multiplication is a cryptographic one-way function, see (18.1).

b) Conversely it is conjectured that factoring $n = pq$ polynomial time reduces to breaking the RSA cryptosystem, implying that these problems are polynomial time equivalent. This is supported by the following: Given $e \in \mathbb{Z}^*_{\varphi(n)}$, being able to compute $d \in \mathbb{Z}^*_{\varphi(n)}$ such that $E_e D_d = \mathrm{id}_{\mathbb{Z}/n\mathbb{Z}}$, allows for the following polynomial time Las-Vegas algorithm to factor $n = pq$:

Let $s \in \mathbb{N}_0$ and $c \in \mathbb{N}$ odd such that $ed - 1 = 2^s c$; note that $2^s \leq ed \leq n^2$ implies that $s \leq 2\log_2(n)$. Hence for $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $1 = x^{ed-1} = (x^c)^{2^s} \in (\mathbb{Z}/n\mathbb{Z})^*$. Thus the order of $x^c \in (\mathbb{Z}/n\mathbb{Z})^*$ is 2^k, for some $k \in \{0, \ldots, s\}$. Now we proceed as follows:

```
choose $x \in \mathbb{Z}_n^*$ randomly
$y \leftarrow (x^c \bmod n)$
for $i \in [0, \ldots, s]$ do
    $g \leftarrow \gcd(y-1, n)$
    if $1 < g < n$ then
        return $g$
    $y \leftarrow (y^2 \bmod n)$
return fail
```

An element $x \in (\mathbb{Z}/n\mathbb{Z})^*$ providing a prime divisor of n is called a **factorisation witness**. We show that the fraction of witnesses is at least $\frac{1}{2}$:

Using the isomorphism $(\mathbb{Z}/n\mathbb{Z})^* \cong (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z})^* \cong C_{p-1} \times C_{q-1}$ shows that the order of $x^c \in (\mathbb{Z}/p\mathbb{Z})^*$ and $x^c \in (\mathbb{Z}/q\mathbb{Z})^*$ is equal to 2^l and 2^m, respectively, where $l, m \in \{0, \ldots, k\}$ and $k = \max\{l, m\}$. Then we have $\gcd((x^c)^{2^i} - 1, n) = p$, say, if and only if $(x^c)^{2^i} = 1 \in (\mathbb{Z}/p\mathbb{Z})^*$ and $(x^c)^{2^i} \neq 1 \in (\mathbb{Z}/q\mathbb{Z})^*$. There is such an $i \in \{0, \ldots, s\}$ if and only if $l < m$. Hence $x \in (\mathbb{Z}/n\mathbb{Z})^*$ is a witness if and only $l \neq m$. We determine the fraction of elements having this property:

Let $p - 1 = 2^t a$ and $q - 1 = 2^u b$, where $t, u \in \mathbb{N}$ and $a, b \in \mathbb{N}$ odd, and let $C_{2^t} \cong G \leq (\mathbb{Z}/p\mathbb{Z})^*$ and $C_{2^u} \cong H \leq (\mathbb{Z}/q\mathbb{Z})^*$ be the uniquely determined subgroups of order 2^t and 2^u, respectively, where we may assume that $t \leq u$. Hence we have $|\{[g,h] \in G \times H; |g| = |h|\}| = \sum_{i=0}^t \varphi(2^i)^2 = 1 + \sum_{i=1}^t (2^{i-1})^2 = 1 + \sum_{i=0}^{t-1} 2^{2i} = 1 + \frac{2^{2t}-1}{2^2-1} = \frac{2^{2t}+2}{3}$.

Since c is odd, and any element of $[g,h] \in G \times H$ is a 2-power, we have $\langle [g^c, h^c] \rangle = \langle [g,h] \rangle$, thus any element of $G \times H$ is a c-th power. Thus the group homomorphism $(\mathbb{Z}/n\mathbb{Z})^* \to G \times H \colon x \mapsto [x^c, x^c]$ is surjective. Hence all elements of $G \times H$ have the same number $\frac{(p-1)(q-1)}{2^t \cdot 2^u} = ab$ of preimages with respect to this map. Thus the fraction of witnesses in $(\mathbb{Z}/n\mathbb{Z})^*$ coincides with the fraction of witnesses in $G \times H$, which by the above is $1 - \frac{2^{2t}+2}{3 \cdot 2^{t+u}} = \frac{3 \cdot 2^{t+u} - 2^{2t} - 2}{3 \cdot 2^{t+u}} \geq \frac{3 \cdot 2^{t+u} - 2^{t+u} - 2}{3 \cdot 2^{t+u}} = \frac{2}{3} - \frac{2}{3 \cdot 2^{t+u}} \geq \frac{2}{3} - \frac{2}{3 \cdot 2^2} = \frac{1}{2}$. $\sharp$

(9.3) Continued fractions. a) Let $r_0 \in \mathbb{N}_0$ and $r_1 \in \mathbb{N}$. Then the Euclidean algorithm yields $l \in \mathbb{N}$ and $q_i \in \mathbb{N}_0$ for $i \in \{1, \ldots, l\}$ such that $r_{i+1} := r_{i-1} - q_i r_i \in \{0, \ldots, r_i - 1\}$ and $r_{l+1} = 0$; hence we have $q_i \geq 1$ for $i \geq 2$. This yields the **(regular) continued fraction**

$$\rho := \frac{r_0}{r_1} = q_1 + \frac{r_2}{r_1} = q_1 + \frac{1}{\frac{r_1}{r_2}} = q_1 + \frac{1}{q_2 + \frac{1}{\frac{r_2}{r_3}}} = \cdots = q_1 + \frac{1}{q_2 + \frac{1}{\ddots + \frac{1}{q_{l-1} + \frac{1}{q_l}}}}.$$

For the latter expression we write $\mathrm{cf}[q_1, \ldots, q_l]$, where more generally we allow for $q_1 \in \mathbb{N}_0$, and $q_i \in \mathbb{N}$ for $i \in \{2, \ldots, l-1\}$, and $1 \leq q_l \in \mathbb{R}$ whenever $l \geq 2$. In particular, if $1 \neq q_l \in \mathbb{N}$ we have $\mathrm{cf}[q_1, \ldots, q_l] = \mathrm{cf}[q_1, \ldots, q_l - 1, 1]$. Then we get a continued fraction expansion for any $\rho \in \mathbb{R}_{\geq 0}$, by taking $q_1 := \lfloor \rho \rfloor \in \mathbb{N}_0$ and proceeding with $\frac{1}{\rho - \lfloor \rho \rfloor}$ instead of ρ; it is of infinite length $l = \infty$ whenever $\rho \notin \mathbb{Q}$.

For $i \in \{1, \ldots, l\}$ let $\rho_i := \frac{\sigma_i}{\tau_i} := \mathrm{cf}[q_1, \ldots, q_i] \in \mathbb{Q}$ be the i-th **convergent**, where $\sigma_i \in \mathbb{N}_0$ and $\tau_i \in \mathbb{N}$ such that $\gcd(\sigma_i, \tau_i) = 1$. Letting additionally $\sigma_{-1} := 0$ and $\tau_{-1} := 1$, as well as $\sigma_0 := 1$ and $\tau_0 := 0$, we by induction on $i \in \{1, \ldots, l\}$ get $\sigma_i = q_i \sigma_{i-1} + \sigma_{i-2}$ and $\tau_i = q_i \tau_{i-1} + \tau_{i-2}$:

We have $\rho_1 = q_1$, yielding $\sigma_1 = q_1 = q_1 \sigma_0 + \sigma_{-1}$ and $\tau_1 = 1 = q_1 \tau_0 + \tau_{-1}$. For $i \geq 2$ we obtain ρ_i from ρ_{i-1} by replacing q_{i-1} by $q_{i-1} + \frac{1}{q_i}$. Thus from $\rho_{i-1} = \frac{\sigma_{i-1}}{\tau_{i-1}} = \frac{q_{i-1}\sigma_{i-2} + \sigma_{i-3}}{q_{i-1}\tau_{i-2} + \tau_{i-3}}$ we get $\rho_i = \frac{(q_{i-1} + \frac{1}{q_i})\sigma_{i-2} + \sigma_{i-3}}{(q_{i-1} + \frac{1}{q_i})\tau_{i-2} + \tau_{i-3}} = \frac{(q_i q_{i-1} + 1)\sigma_{i-2} + q_i \sigma_{i-3}}{(q_i q_{i-1} + 1)\tau_{i-2} + q_i \tau_{i-3}} = \frac{q_i \sigma_{i-1} + \sigma_{i-2}}{q_i \tau_{i-1} + \tau_{i-2}}$. For $i \in \{0, \ldots, l\}$ we moreover have $\sigma_i \tau_{i-1} - \sigma_{i-1} \tau_i = (-1)^i$, implying that indeed $\gcd(\sigma_i, \tau_i) = 1$: We have $\sigma_0 \tau_{-1} - \sigma_{-1} \tau_0 = 1$ and for $i \geq 1$ we obtain $\sigma_i \tau_{i-1} - \sigma_{i-1} \tau_i = (q_i \sigma_{i-1} + \sigma_{i-2}) \tau_{i-1} - \sigma_{i-1}(q_i \tau_{i-1} + \tau_{i-2}) = -(\sigma_{i-1} \tau_{i-2} - \sigma_{i-2} \tau_{i-1}) = -(-1)^{i-1} = (-1)^i$. $\sharp$

For $i \in \{1, \ldots, l-1\}$ the ρ_i are approximations of ρ in the following sense: Let $\omega_i := \mathrm{cf}[q_{i+1}, q_{i+2}, \ldots] \in \mathbb{R}$, hence we have $\rho = \mathrm{cf}[q_1, \ldots, q_i, \omega_i]$ and $\omega_i \geq q_{i+1} > 0$. Thus ρ is obtained from ρ_i by replacing q_i by $q_i + \frac{1}{\omega_i}$. This implies $\rho = \frac{(q_i + \frac{1}{\omega_i})\sigma_{i-1} + \sigma_{i-2}}{(q_i + \frac{1}{\omega_i})\tau_{i-1} + \tau_{i-2}} = \frac{\omega_i \sigma_i + \sigma_{i-1}}{\omega_i \tau_i + \tau_{i-1}}$, thus since $\tau_{i+1} > \tau_i$ we get $|\rho - \rho_i| = |\frac{\omega_i \sigma_i + \sigma_{i-1}}{\omega_i \tau_i + \tau_{i-1}} - \frac{\sigma_i}{\tau_i}| = |\frac{\sigma_{i-1}\tau_i - \sigma_i \tau_{i-1}}{(\omega_i \tau_i + \tau_{i-1})\tau_i}| = \frac{1}{(\omega_i \tau_i + \tau_{i-1})\tau_i} \leq \frac{1}{(q_{i+1}\tau_i + \tau_{i-1})\tau_i} = \frac{1}{\tau_{i+1}\tau_i} < \frac{1}{\tau_i^2}$.

b) Let $\gamma \in \mathbb{R}_{\geq 0}$, and let $x \in \mathbb{N}_0$ and $y \in \mathbb{N}$ such that $|\gamma - \frac{x}{y}| \leq \frac{1}{2y^2}$, where we may assume that $\gcd(x, y) = 1$; hence $\frac{x}{y} \in \mathbb{Q}$ approximates γ somewhat better than guaranteed by the bound for its convergents. Then we have **Legendre's Theorem**: The fraction $\frac{x}{y} \in \mathbb{Q}$ indeed is a convergent:

Let $\frac{x}{y} = \mathrm{cf}[q_1, \ldots, q_l]$ be the continued fraction expansion of $\frac{x}{y}$, and let $\rho_i = \frac{\sigma_i}{\tau_i}$, for $i \in \{1, \ldots, l\}$, be the convergents of $\frac{x}{y}$, hence we have $x = \sigma_l$ and $y = \tau_l$. By assumption there are $\epsilon \in \{\pm 1\}$ and $\delta \in \mathbb{R}$ such that $0 \leq \delta \leq \frac{1}{2}$ and $\gamma = \frac{\sigma_l}{\tau_l} + \frac{\epsilon\delta}{\tau_l^2}$, where we may assume that $\delta > 0$ and $\epsilon = (-1)^{l-1}$. Let $\omega := \frac{\tau_l - \delta\tau_{l-1}}{\delta\tau_l} \in \mathbb{R}$. Since $\tau_{l-1} < \tau_l$ we have $(1-\delta)\tau_l \geq \frac{\tau_l}{2} > \frac{\tau_{l-1}}{2} \geq \delta\tau_{l-1}$, implying $\tau_l - \delta\tau_{l-1} > \delta\tau_l$, and thus $\omega = \frac{\tau_l - \delta\tau_{l-1}}{\delta\tau_l} > 1$. We have $\mathrm{cf}[q_1, \ldots, q_l, \omega] = \frac{(q_l + \frac{1}{\omega})\sigma_{l-1} + \sigma_{l-2}}{(q_l + \frac{1}{\omega})\tau_{l-1} + \tau_{l-2}} = \frac{\omega\sigma_l + \sigma_{l-1}}{\omega\tau_l + \tau_{l-1}} = \frac{(\frac{\tau_l - \delta\tau_{l-1}}{\delta\tau_l})\sigma_l + \sigma_{l-1}}{(\frac{\tau_l - \delta\tau_{l-1}}{\delta\tau_l})\tau_l + \tau_{l-1}} = \frac{\sigma_l\tau_l - \delta(\sigma_l\tau_{l-1} - \sigma_{l-1}\tau_l)}{\tau_l^2} = \frac{\sigma_l}{\tau_l} - \frac{(-1)^l\delta}{\tau_l^2} = \gamma$. Thus replacing ω by its continued fraction expansion, which starts with a positive integer, yields the expansion of γ as a prolongation of $\mathrm{cf}[q_1, \ldots, q_l]$. $\sharp$

(9.4) Low decryption exponent attack [Wiener, 1990]. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$. Let $e, d \in \mathbb{Z}_{\varphi(n)}$ such that $d \neq 1$ and $ed = 1 \in \mathbb{Z}/\varphi(n)\mathbb{Z}$. There is $k \in \mathbb{N}$ such that $ed - k\varphi(n) = 1$, hence we have $\gcd(k, d) = 1$ and from $k\varphi(n) = ed - 1 < d\varphi(n)$ we infer $k < d$. Thus, if we are able to determine $\frac{k}{d} \in \mathbb{Q}$, then d is found, and the RSA cryptosystem is broken.

Assume that $q < p < \lambda q$ for some $\lambda > 1$, and that $d < \frac{\sqrt[4]{n}}{\sqrt{2(\lambda+1)}}$. Hence we have $0 < n - \varphi(n) = pq - (p-1)(q-1) = p + q - 1 < (\lambda+1)q < (\lambda+1)\sqrt{n}$. This yields $|\frac{e}{n} - \frac{k}{d}| = |\frac{ed - kn}{dn}| = |\frac{ed - k(n - \varphi(n)) - k\varphi(n)}{dn}| = |\frac{1 - k(n - \varphi(n))}{dn}| < \frac{k(n - \varphi(n))}{dn} < \frac{d(\lambda+1)\sqrt{n}}{dn} = \frac{\lambda+1}{\sqrt{n}} < \frac{\lambda+1}{2(\lambda+1)d^2} = \frac{1}{2d^2}$. Hence $\frac{k}{d} \in \mathbb{Q}$ is a convergent of $\rho := \frac{e}{n} \in \mathbb{Q}$, and can be computed in polynomial time by the Euclidean algorithm. For the convergents $\rho_i = \frac{\sigma_i}{\tau_i} \in \mathbb{Q}$, where $\sigma_i, \tau_i \in \mathbb{N}$ such that $\gcd(\sigma_i, \tau_i) = 1$, letting $\psi_i := \frac{e\tau_i - 1}{\sigma_i} \in \mathbb{Q}$, we solve the quadratic equation $X^2 + (\psi_i - n - 1)X + n = 0$, yielding p and q if the convergent taken was the right one.

(9.5) Practical aspects of the RSA cryptosystem. a) Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$.

Given the capabilities of the known factorisation algorithms, p and q should be chosen of the same size, which nowadays should be at least $\sim 2^{512} \sim 10^{154}$, thus $n \sim 2^{1024} \sim 10^{308}$. If there are factorisation algorithms running faster for certain choices of the prime divisors of n, for example if $p - 1$ or $p + 1$ has no large prime divisors, then these have to be avoided as well. Moreover, $|p - q|$ must not be too small, since otherwise $p, q \sim \sqrt{n} \in \mathbb{R}$ and the prime divisors of n can be found by trial division with integers close to $\sqrt{n}$.

The state of the art, as far as integer factorisation is concerned, is reflected by the **RSA challenge** problems. The current record [Kleinjung et al., 2009] is the factorisation of the modulus 'RSA-768', which is of size $\sim 2^{768} \sim 10^{232}$, by a parallel computing approach which needed an equivalent of some 2000 years of serial computing time. (A few smaller ones have been factored since then.)

To give an explicit example, we present the details of the modulus 'RSA-100', which is of size $\sim 2^{330} \sim 10^{100}$, and has been factored [A. Lenstra, 1991] using the **multipolynomial quadratic sieve (MPQS)**, which nowadays would need roughly one hour of computing time:

$$\begin{aligned} n &:= 1522605027922533360535618378132637429718068114961 3\\ & 8068865790849458012296325895289765400035069200613 9\\ p &:= 37975227936943673922808872755445627854565536638199\\ q &:= 40094690950920881030683735292761468389214899724061 \end{aligned}$$

The factorisation of $p-1$ and $q-1$ is easily found (in less than a second of computing time):

$$\begin{aligned} p-1 &= 2 \cdot 3167 \cdot 3613 \cdot 587546788471 \cdot 3263521422991 \cdot 865417043661324529\\ q-1 &= 2^2 \cdot 5 \cdot 41 \cdot 2119363 \cdot 602799725049211 \cdot 38273186726790856290328531 \end{aligned}$$

In particular, we have $\gcd(p-1, q-1) = 2$. Moreover, $p-1$ and $q-1$ indeed have large prime divisors, and letting $p' := 865417043661324529$ and $q' := 38273186726790856290328531$ be the respective largest ones in turn we get:

$$\begin{aligned} p'-1 &= 2^4 \cdot 3 \cdot 11 \cdot 17 \cdot 61 \cdot 1580566471723\\ q'-1 &= 2 \cdot 3 \cdot 5 \cdot 61 \cdot 113 \cdot 93557 \cdot 1978284752702551 \end{aligned}$$

In particular, $p'-1$ and $q'-1$ have large prime divisors as well.

b) In practice, the **Public-Key Cryptography Standard (PKCS) #1** is used, for example in the **SSL/TLS Handshake Protocol**; this is about 10^3 times slower than DES or AES. Randomisation is done using **Optimal Asymmetric Encryption Padding (OAEP)**, which runs as follows:

Let $b := \lceil \log_2(n) \rceil \in \mathbb{N}$ be the number of binary digits needed to represent n. Moreover, let $k, l \in \mathbb{N}$ such that $k + l = b - 1$. Then the elements of $\mathbb{Z}_2^{k+l}$ can be considered as binary representations of integers in $\mathbb{Z}_{2^{k+l}} \subseteq \mathbb{Z}_n$. Now let $\mathbb{Z}_2^l$ be the set of plaintexts of a block cipher of length l over $\mathbb{Z}_2$, and let $\mathbb{Z}_2^k$ be a source of random vectors. Finally, let $f\colon \mathbb{Z}_2^k \to \mathbb{Z}_2^l$ and $g\colon \mathbb{Z}_2^l \to \mathbb{Z}_2^k$ be 'random' **expansion** and **compression** functions, respectively, which are made public:

Then encryption runs as follows: Let $v \in \mathbb{Z}_2^l$ be a plaintext, let $u \in \mathbb{Z}_2^k$ be randomly chosen, and let $w := \big((f(u) \oplus v) \cdot 2^k\big) \oplus \big(u \oplus g(f(u) \oplus v)\big) \in \mathbb{Z}_2^{k+l} \subseteq \mathbb{Z}_n$. Hence both v and u are masked: The former by adding $f(u)$, which since u is chosen randomly is random as well; the latter by adding $g(f(u) \oplus v)$, which since $f(u) \oplus v$ can be considered random is random as well. Both ingredients are incorporated into w, which is encrypted yielding $\widehat{w} := (w^e \bmod n) \in \mathbb{Z}_n$.

Decryption runs as follows: We recover $w := (\widehat{w}^d \bmod n) \in \mathbb{Z}_2^{k+l} \subseteq \mathbb{Z}_n$. Then we have $(w \div 2^k) = f(u) \oplus v \in \mathbb{Z}_2^l$ and $(w \bmod 2^k) = u \oplus g(f(u) \oplus v) \in \mathbb{Z}_2^k$, hence we recover $(w \bmod 2^k) \oplus g(w \div 2^k) = \big(u \oplus g(f(u) \oplus v)\big) \oplus g(f(u) \oplus v) = u \in \mathbb{Z}_2^k$, and subsequently $f(u) \oplus (w \div 2^k) = f(u) \oplus (f(u) \oplus v) = v \in \mathbb{Z}_2^l$. $\sharp$

In the above, the functions f and g are considered to obey to the **random oracle model** [Bellare-Rogaway, 1993], which says that they are evaluated by an

oracle, rather than by a formula, for example; hence no information on preimages can be obtained from images. But k has to be chosen large enough such that a complete search through $\mathbb{Z}_2^k$ is infeasible: Otherwise, we would be able to compute all possible encryptions of a given plaintext, rendering randomisation useless. This is a drawback, since it reduces the fraction of information bits to $\frac{l}{b}$; typical values nowadays are $k+l+1=b=1024$ and $k=128$.

10 The Rabin cryptosystem

(10.1) The Rabin cryptosystem [1979]. Let $p \neq q \in \mathbb{N}$ be primes such that $p, q \equiv 3 \pmod 4$, and let $n := pq \in \mathbb{N}$ be the modulus. Let $\mathcal{P} = \mathcal{C} := \mathbb{Z}/n\mathbb{Z}$, and let the only encryption function be $E \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \colon x \mapsto x^2$. The public key is n, the private key is $[p,q]$.

Since $2 \in (\mathbb{Z}/\varphi(n)\mathbb{Z}) \setminus (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, decryption is not unique, thus this is not a cryptosystem. For $x \in \mathbb{Z}/n\mathbb{Z}$ there are at most four square roots of x^2 in $\mathbb{Z}/n\mathbb{Z}$, and if p and q are known these can be computed in polynomial time as follows:

For any $y \in \mathbb{Z}/p\mathbb{Z}$ the polynomial $X^2 - y \in \mathbb{Z}/p\mathbb{Z}[X]$ has at most two roots in $\mathbb{Z}/p\mathbb{Z}$, and $y_p \in \mathbb{Z}/p\mathbb{Z}$ is a root if and only if $-y_p \in \mathbb{Z}/p\mathbb{Z}$ is a root as well; we have $y_p \neq -y_p$ if and only if $y_p \neq 0$, which holds if and only if $y \neq 0$. If $X^2 - y$ has a root $x \in \mathbb{Z}/p\mathbb{Z}$ at all, i. e. we have $y = x^2$, then for $y_p := y^{\frac{p+1}{4}} \in \mathbb{Z}/p\mathbb{Z}$ we have $y_p^2 = (y^{\frac{p+1}{4}})^2 = ((x^2)^{\frac{p+1}{4}})^2 = x^{p+1} = x^{(p-1)+2} = x^2 = y \in \mathbb{Z}/p\mathbb{Z}$, and thus $\{\pm y_p\}$ indeed are the roots of $X^2 - y$ in $\mathbb{Z}/p\mathbb{Z}$. If $X^2 - y$ does not have a root in $\mathbb{Z}/p\mathbb{Z}$, then this is detected by observing that $y_p^2 \neq y \in \mathbb{Z}/p\mathbb{Z}$.

Given $y \in \mathbb{Z}/n\mathbb{Z}$, using the ring isomorphism $\mathbb{Z}/n\mathbb{Z} \to (\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ we conclude that $X^2 - y \in \mathbb{Z}/n\mathbb{Z}[X]$ has a root in $\mathbb{Z}/n\mathbb{Z}$ if and only if both $X^2 - y \in \mathbb{Z}/p\mathbb{Z}[X]$ and $X^2 - y \in \mathbb{Z}/q\mathbb{Z}[X]$ have a root in $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/q\mathbb{Z}$, respectively. In this case there are at most four roots in $\mathbb{Z}/n\mathbb{Z}$, given as the preimages of $[\pm y_p, \pm y_q] \in (\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$, where $y_p \in \mathbb{Z}_p$ and $y_q \in \mathbb{Z}_q$ are such that $y_p = y^{\frac{p+1}{4}} \in \mathbb{Z}/p\mathbb{Z}$ and $y_q = y^{\frac{q+1}{4}} \in \mathbb{Z}/q\mathbb{Z}$. Letting $s_p, s_q \in \mathbb{Z}$ such that $1 = s_p p + s_q q$, and $x \in \mathbb{Z}_n$ such that $x = (\pm y_p s_q q) + (\pm y_q s_p p) \in \mathbb{Z}/n\mathbb{Z}$, we have $x = \pm y_p(1 - s_p p) = \pm y_p \in \mathbb{Z}/p\mathbb{Z}$ and $x = \pm y_q(1 - s_q q) = \pm y_q \in \mathbb{Z}/q\mathbb{Z}$, and thus $x^2 = y \in \mathbb{Z}/n\mathbb{Z}$. ♯

Rabin encryption is faster than RSA encryption, while Rabin decryption is about as fast as RSA decryption. Similar to the RSA cryptosystem, the Rabin cryptosystem is vulnerable to protocol failure by multiplicativity or low exponent attacks, countermeasures again are restricting plaintexts to admissible subsets and randomisation, respectively. Since decryption of a ciphertext yields up to four possible plaintexts, from which the correct one has to be determined, plaintexts should be chosen from an admissible subset anyway:

E. g. let $p := 251$ and $q := 263$, hence $n = 66013$ and $2^{16} = 65536 < n$. We consider a **Rabin block cipher**: We encode elements of $\mathbb{Z}_2^{10}$ into $\mathbb{Z}_2^{16}$ by repeating the last 6 letters, and the latter are considered via 2-adic expansion as elements of $\mathbb{Z}_{2^{16}} \subseteq \mathbb{Z}_n$. E. g. `1001111001` $\in \mathbb{Z}_2^{10}$ is encoded into

$x := \texttt{1001111001|111001} \in \mathbb{Z}_2^{16}$, yielding $x = 40569 \in \mathbb{Z}_n$. Thus $y = x^2 \in \mathbb{Z}/n\mathbb{Z}$ where $y := 7645 \in \mathbb{Z}_n$, and $y_p = 93 \in \mathbb{Z}_p$ and $y_q = 196 \in \mathbb{Z}_q$. Using $s_p = -22$ and $s_q = 21$ we get the following square roots, only x_4 being admissible:

	$\mathbb{Z}_n$	$\mathbb{Z}_2^{16}$
x_1	25444	0010011011|000110
x_2	54374	0110011000|101011
x_3	11639	1110111010|110100
x_4	40569	1001111001|111001

(10.2) Protocol failure of the Rabin cryptosystem. a) By the above, protocol failure of the Rabin cryptosystem polynomial time reduces to factoring n. Conversely, protocol failure of the Rabin cryptosystem, i. e. having an oracle being able to compute a square root for any square in $(\mathbb{Z}/n\mathbb{Z})^*$, yields the following polynomial time Las-Vegas algorithm to factor n:

```
choose $x \in \mathbb{Z}_n$ randomly
if $x = 0$ then return fail
$g := \gcd(x, n) \in \mathbb{Z}_n$
if $1 < g < n$ then return g
compute $x' \in \mathbb{Z}_n$ such that $(x')^2 = x^2 \in \mathbb{Z}/n\mathbb{Z}$
$g := \gcd(x - x', n) \in \mathbb{Z}_n$
if $1 < g < n$ then return g
return fail
```

For $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $x \neq -x \in \mathbb{Z}/p\mathbb{Z}$ and $x \neq -x \in \mathbb{Z}/q\mathbb{Z}$, hence we have precisely four cases $x' = \pm x \in \mathbb{Z}/p\mathbb{Z}$ and $x' = \pm x \in \mathbb{Z}/q\mathbb{Z}$. The cases with equal signs yield $\gcd(x - x', n) = \gcd(0, n) = n$ and $\gcd(x - x', n) = \gcd(2x, n) = 1$, respectively, while the cases with different signs yield $\gcd(x - x', n) = p$ and $\gcd(x - x', n) = q$, respectively. Thus in the latter two cases x is a factorisation witness, while in the former two cases it is not. Hence the fraction of witnesses is precisely $\frac{n-1-\varphi(n)+\frac{\varphi(n)}{2}}{n} = \frac{1}{2} + \frac{p+q-3}{2pq} > \frac{1}{2}$. ♯

Hence the Rabin cryptosystem is provably secure, relative to factoring the modulus n, against a chosen-plaintext attack: Given a plaintext $x \in \mathbb{Z}/n\mathbb{Z}$, any decryption $x' \in \mathbb{Z}/n\mathbb{Z}$ of the ciphertext $x^2 \in \mathbb{Z}/n\mathbb{Z}$, which by assumption can be found, with probability more than $\frac{1}{2}$ leads to a factorisation of n. It is conjectured that squaring in $\mathbb{Z}/n\mathbb{Z}$ is a cryptographic one-way function, see (18.1).

b) At the same time, the Rabin cryptosystem is insecure against a chosen-ciphertext attack, inasmuch in this case the hypothetical decryption algorithm in a chosen-plaintext attack is just replaced by an actual decryption algorithm.

This is prevented if only plaintexts from an admissible subset, e. g. as described above, are allowed: If $x \in \mathbb{Z}/n\mathbb{Z}$ is such that no square root of $x^2 \in \mathbb{Z}/n\mathbb{Z}$ is admissible, then decryption of x^2 simply is forbidden. However, in this case we can no longer prove that protocol failure necessarily leads to a factorisation algorithm for n, i. e. we no longer have provable security.

11 The ElGamal cryptosystem

(11.1) The ElGamal cryptosystem [1985]. Let $p \in \mathbb{N}$ be a prime, and let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root. Hence given $\beta \in (\mathbb{Z}/p\mathbb{Z})^*$ there is a unique **discrete logarithm** $b = \log_\rho(\beta) \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $\beta = \rho^b$.

Let $\mathcal{P} := (\mathbb{Z}/p\mathbb{Z})^*$ and $\mathcal{C} := (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^*$; ciphertexts being twice as long as plaintexts is a disadvantageous **message expansion**. For $[a,b] \in \mathcal{K} := (\mathbb{Z}/(p-1)\mathbb{Z}) \times (\mathbb{Z}/(p-1)\mathbb{Z})$ let $\alpha := \rho^a \in (\mathbb{Z}/p\mathbb{Z})^*$, the public key is $[p,\rho,\alpha]$, while a is private to Alice, and b is private to Bob. To send messages from Bob to Alice, let $E_b \colon (\mathbb{Z}/p\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^* \colon x \mapsto [\rho^b, x\alpha^b]$, and $D_a \colon (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^* \colon [\beta,y] \mapsto y\beta^{-a}$. Then we have $D_a(E_b(x)) = D_a(\rho^b, x\alpha^b) = x\alpha^b(\rho^b)^{-a} = x(\rho^a)^b(\rho^b)^{-a} = x \in (\mathbb{Z}/p\mathbb{Z})^*$.

The ElGamal cryptosystem is used as a randomised cryptosystem: Let $\mu_\mathcal{P}$ be a probability distribution on $\mathcal{P}$. Since $\rho^b \in (\mathbb{Z}/p\mathbb{Z})^*$ is a primitive root if and only if $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, for $\mathcal{C}' := \{\beta \in (\mathbb{Z}/p\mathbb{Z})^*; \beta \text{ primitive root}\} \times (\mathbb{Z}/p\mathbb{Z})^* \subseteq \mathcal{C}$ and $[a,b] \in \mathcal{K}' := (\mathbb{Z}/(p-1)\mathbb{Z}) \times (\mathbb{Z}/(p-1)\mathbb{Z})^*$ we get $E_b \colon \mathcal{P} \to \mathcal{C}'$; note that $|\mathcal{C}'| = \varphi(p-1)\cdot(p-1) = |\mathcal{K}'|$. Let $\mu_{\mathcal{K}'}$ be the uniform distribution, i. e. the components of $\mathcal{K}'$ are uniformly distributed and independent, and assume that $\mu_{\mathcal{K}'}$ and $\mu_\mathcal{P}$ are independent. Then $\mu_{\mathcal{C}'}$ is the uniform distribution and the ElGamal cryptosystem is perfectly secure, i. e. from observing ciphertexts alone, neglecting the fact that α is publicly known, no information about plaintexts can be extracted:

For $[\beta,y] \in \mathcal{C}'$ we have $E_b(x) = [\beta,y]$, where $[x;a,b] \in \mathcal{P} \times \mathcal{K}'$, if and only if $\beta = \rho^b$ and $y = x\alpha^b = x\rho^{ab} = x(\rho^b)^a \in (\mathbb{Z}/p\mathbb{Z})^*$. Thus $b = \log_\rho(\beta) \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ is uniquely determined. Moreover, for any $x \in (\mathbb{Z}/p\mathbb{Z})^*$ we get $a = \log_{\rho^b}(yx^{-1}) = b^{-1}\cdot\log_\rho(yx^{-1}) \in \mathbb{Z}/(p-1)\mathbb{Z}$, in other words a is uniquely determined as soon as b and x are given; note that here we use $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$. Hence we have $\mu_{\mathcal{C}'}(\beta,y) = \frac{1}{\varphi(p-1)}\cdot\sum_{x\in\mathcal{P}}\frac{\mu_\mathcal{P}(x)}{p-1} = \frac{1}{(p-1)\cdot\varphi(p-1)} = \frac{1}{|\mathcal{C}'|}$. Moreover, for any $[\beta,y] \in \mathcal{C}'$ and $x \in \mathcal{P}$ this shows $\mu_{\mathcal{P}\times\mathcal{C}'}(x;\beta,y) = \frac{\mu_\mathcal{P}(x)}{|\mathcal{K}'|} = \frac{\mu_\mathcal{P}(x)}{|\mathcal{C}'|} = \mu_\mathcal{P}(x)\cdot\mu_{\mathcal{C}'}(\beta,y)$, thus $\mu_\mathcal{P}(x \mid \beta,y) = \frac{\mu_{\mathcal{P}\times\mathcal{C}'}(x;\beta,y)}{\mu_{\mathcal{C}'}(\beta,y)} = \mu_\mathcal{P}(x)$. ♯

(11.2) Protocol failure of the unrandomised ElGamal cryptosystem. We keep the setting of (11.1). To prevent the following attacks, the second component of the key used has to be varied for each encryption.

a) Malleability. Changing ciphertexts suitably leads to a controllable change of the associated plaintext: Given $[a,b] \in \mathcal{K}$, a plaintext $x \in (\mathbb{Z}/p\mathbb{Z})^*$ and the associated ciphertext $[\rho^b, x\alpha^b] \in \mathcal{C}$, then for any $x' \in (\mathbb{Z}/p\mathbb{Z})^*$ the encryption of $x'x$ is $[\rho^b, x'(x\alpha^b)] \in \mathcal{C}$, hence can be computed from the encryption of x.

b) There is the following protocol failure: Let $x, x' \in (\mathbb{Z}/p\mathbb{Z})^*$ be encrypted with $[a,b] \in \mathcal{K}$, yielding ciphertexts $[\rho^b, y]$ and $[\rho^b, y']$, respectively, where $y := x\alpha^b \in (\mathbb{Z}/p\mathbb{Z})^*$ and $y' := x'\alpha^b \in (\mathbb{Z}/p\mathbb{Z})^*$. Hence $y^{-1}y' = x^{-1}x' \in (\mathbb{Z}/p\mathbb{Z})^*$, and thus the plaintext $x = yy'^{-1}x' \in (\mathbb{Z}/p\mathbb{Z})^*$ can be computed from the plaintext x' and the ciphertexts y and y'.

(11.3) Breaking the ElGamal cryptosystem. We keep the setting of (11.1). Breaking the ElGamal cryptosystem, by finding the first component $a \in \mathbb{Z}/(p-1)\mathbb{Z}$ of the key used, polynomial time reduces to solving the **discrete logarithm problem** in $(\mathbb{Z}/p\mathbb{Z})^*$: Given a primitive root $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ and $\alpha \in (\mathbb{Z}/p\mathbb{Z})^*$, find the discrete logarithm $a = \log_\rho(\alpha) \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $\rho^a = \alpha$. Thus this cryptosystem is secure only if solving the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$ is computationally difficult; it is conjectured that exponentiation in $(\mathbb{Z}/p\mathbb{Z})^*$ is a cryptographic one-way function, see (18.1).

Conversely, it is conjectured that solving the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$ polynomial time reduces to breaking the ElGamal cryptosystem. Given the capabilities of modern algorithms for the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$, the prime p should be of size $p \sim 2^{1024} \sim 10^{308}$, and $p-1$ should have a large prime divisor.

(11.4) The Diffie-Hellman key exchange protocol [1976]. The original idea behind the ElGamal cryptosystem is the following protocol, which actually was the very begin of public-key cryptography:

a) Alice and Bob want to agree on a common private key, using an insecure channel. To do so, Alice and Bob publicly agree on a prime $p \in \mathbb{N}$ and a primitive root $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$. Then Alice picks $a \in \mathbb{Z}/(p-1)\mathbb{Z}$ and sends $\alpha := \rho^a \in (\mathbb{Z}/p\mathbb{Z})^*$ to Bob, similarly Bob picks $b \in \mathbb{Z}/(p-1)\mathbb{Z}$ and sends $\beta := \rho^b \in (\mathbb{Z}/p\mathbb{Z})^*$ to Alice. Finally, Alice computes $\gamma := \beta^a = \rho^{ab} \in (\mathbb{Z}/p\mathbb{Z})^*$, Bob computes $\gamma := \alpha^b = \rho^{ab} \in (\mathbb{Z}/p\mathbb{Z})^*$, hence γ is their common private key.

It is not necessary to choose ρ as a primitive root, but just to choose $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ having a sufficiently large order. To compute the order of ρ efficiently, we have to be able to factor $p-1$.

A **man-in-the-middle attack** leads to the following protocol failure: Since Alice and Bob cannot verify who has sent the messages, an adversary Oscar can impersonate Alice and exchange a key with Bob, and Oscar can impersonate Bob and exchange a key with Alice, and then Oscar is able to intercept all messages. This attack is prevented using **signatures**.

b) Determining the private key $\gamma \in (\mathbb{Z}/p\mathbb{Z})^*$ is called the **Diffie-Hellman problem**: Given a prime $p \in \mathbb{N}$, a primitive root $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$, and $\alpha := \rho^a \in (\mathbb{Z}/p\mathbb{Z})^*$ as well as $\beta := \rho^b \in (\mathbb{Z}/p\mathbb{Z})^*$, without knowing $a, b \in \mathbb{Z}/(p-1)\mathbb{Z}$ find $\gamma := \rho^{ab} \in (\mathbb{Z}/p\mathbb{Z})^*$. Hence solving the Diffie-Hellman problem polynomial time reduces to solving the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$. Conversely, it is an open problem whether solving the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$ polynomial time reduces to solving the Diffie-Hellman problem.

Solving the Diffie-Hellman problem is polynomial time equivalent to protocol failure of the ElGamal cryptosystem: Assume we have a Diffie-Hellman oracle, and let $[\beta, y] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^*$ be an ElGamal ciphertext, keeping the setting of (11.1). Using the oracle with input $[\rho, \alpha, \beta]$ compute $\gamma \in (\mathbb{Z}/p\mathbb{Z})^*$, then $y\gamma^{-1} = x\alpha^b\rho^{-ab} = x(\rho^a)^b\rho^{-ab} = x \in (\mathbb{Z}/p\mathbb{Z})^*$ is the plaintext associated with $[\beta, y]$. Conversely, assume we have an ElGamal decryption oracle, and let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root and $\alpha, \beta \in (\mathbb{Z}/p\mathbb{Z})^*$. Using the oracle with input $[\beta, 1] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^*$ compute $x \in (\mathbb{Z}/p\mathbb{Z})^*$, then $1 = x\alpha^b = x\rho^{ab}$ implies $x^{-1} = \gamma \in (\mathbb{Z}/p\mathbb{Z})^*$.

Since breaking the ElGamal cryptosystem polynomial time reduces to solving the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$, and conjecturally is polynomial time equivalent to it, this indicates that the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$ in general might not polynomial time reduce to solving the Diffie-Hellman problem. This changes if $p-1$ has only small prime factors, see [10, Ch.3.7].

(11.5) Generalised ElGamal cryptosystems. For any finite cyclic group $G = \langle \rho \rangle$ of known order $|G|$, there is an associated discrete logarithm problem: Given $\gamma \in G$, find the discrete logarithm $c = \log_\rho(\gamma) \in \mathbb{Z}/|G|\mathbb{Z}$ such that $\gamma = \rho^c$. Any such discrete logarithm problem yields a **generalised ElGamal cryptosystem**, which is secure only if solving the underlying discrete logarithm problem is computationally difficult.

There are discrete logarithm problems which can be solved in polynomial time; e. g. for $n \in \mathbb{N}$ the additive group $\mathbb{Z}/n\mathbb{Z}$ is cyclic with generator $a \in (\mathbb{Z}/n\mathbb{Z})^*$, and for $x \in \mathbb{Z}/n\mathbb{Z}$ we have $\log_a(x) = xa^{-1} \in \mathbb{Z}/n\mathbb{Z}$, which hence can be computed in polynomial time by the Euclidean algorithm. Still, there are finite commutative groups possessing cyclic subgroups having conjecturally difficult discrete logarithm problems:

a) $(\mathbb{Z}/p\mathbb{Z})^*$ where $p \in \mathbb{N}$ is a prime, and $(\mathbb{Z}/n\mathbb{Z})^*$ where $n \in \mathbb{N}$ composite.
b) $\mathbb{F}_q^*$, where $\mathbb{F}_q$ is the field with q elements.
c) The points of an **elliptic curve** over $\mathbb{F}_q$ [Miller, 1986; Koblitz, 1987; Menezes-Vanstone, 1993], see [8, Ch.6].
d) The points of the **Jacobian variety** of an **hyperelliptic curve** over $\mathbb{F}_q$ [Koblitz, 1989; Frey, ≥1994], see [8, Ch.6].
e) Class groups of algebraic number fields [Buchmann, 1990].

(11.6) Randomised public key cryptosystems. Next to the generalised ElGamal cryptosystems there are other randomised public key cryptosystems: The **Blum-Goldwasser cryptosystem** [1985], which is the most efficient randomised public key cryptosystem and comparable to the RSA cryptosystem, is based on the computational difficulty of integer factorisation, see [10, Ch.8.7.2].

The **Goldwasser-Micali cryptosystem** [1982] is based on the computational difficulty of the **quadratic residuosity** problem for integers, see [10, Ch.8.7.1]; see (18.1) for the associated function problem of taking integer modular square roots, and (28.1) for the case of prime moduli.

The **McEliece cryptosystem** [1978], which was the first realisation of a randomised public key cryptosystem and is very efficient, but has due to its large public keys received little practical attention, is based on the computational difficulty of **decoding linear error-correcting codes**, which is an NP-hard problem, see [10, Ch.8.5]. Similar to the Merkle-Hellman cryptosystem, see (12.2), which is based on the NP-hard subset sum problem, the idea is to consider particular codes having polynomial time decoding algorithms, e. g. **algebraic geometric codes** such as **Goppa codes**, and to disguise them; the version using Goppa codes is still unbroken.

12 Knapsack cryptosystems

(12.1) The subset sum problem. Given $s \in \mathbb{N}_0$, as well as $n \in \mathbb{N}$ and a **knapsack sequence** $[a_1, \ldots, a_n] \in \mathbb{N}_0^n$, the **subset sum problem** is to find a sequence $[x_1, \ldots, x_n] \in \{0,1\}^n$ such that $\sum_{i=1}^n x_i a_i = s$, or to show that such a sequence does not exist; in general a solution sequence is not necessarily unique.

The subset sum problem is solved by the following recursive algorithm S, returning $S(s; a_1, \ldots, a_n)$, which is a solution sequence if such a sequence exists, or fail otherwise:

```
if $n = 1$ then
    if $s = 0$ then return $[0]$
    if $s = a_1$ then return $[1]$
    return fail
$T := S(s; a_1, \ldots, a_{n-1})$        # recursion
if $T \neq$ fail then return $[T, 0]$
if $s \geq a_n$ then
    $T := S(s - a_n; a_1, \ldots, a_{n-1})$        # recursion
    if $T \neq$ fail then return $[T, 1]$
return fail
```

Since S is recursively called twice for given n, for the runtime we have $t(n) = 2t(n-1)+1$ for $n \geq 2$ and $t(1) = 1$. This by induction yields $t(n) = 2^n - 1$, hence S has exponential running time; actually S runs in non-deterministic linear time. No polynomial time deterministic algorithm to solve the subset sum problem is known, actually this problem is NP-hard. The following particular case can be easily solved:

A knapsack sequence $[a_1, \ldots, a_n] \in \mathbb{N}^n$ is called **superincreasing** if $a_j > \sum_{i=1}^{j-1} a_i$ for $j \in \{1, \ldots, n\}$; hence if there is a solution sequence it is unique. The subset sum problem for superincreasing sequences is solved by the following recursive algorithm S', returning $S'(s; a_1, \ldots, a_n)$, which is the solution sequence if it exists, or fail otherwise; since S' is recursively called only once for given n, it has linear running time:

```
if $n = 1$ then
    if $s = 0$ then return $[0]$
    if $s = a_1$ then return $[1]$
    return fail
if $s \geq a_n$ then
    $T' := S'(s - a_n; a_1, \ldots, a_{n-1})$        # recursion
```

```
        if $T' \neq$ fail then return $[T', 1]$
    else
        $T' := S'(s; a_1, \ldots, a_{n-1})$        # recursion
        if $T' \neq$ fail then return $[T', 0]$
    return fail
```

(12.2) The Merkle-Hellman cryptosystem [1978]. It is based on the subset sum problem, and was the first realisation of a public-key cryptosystem: Let $n, m \in \mathbb{N}$. For $c \in \mathbb{Z}_m^*$ and $a := [a_1, \ldots, a_n] \in \mathbb{N}^n$ superincreasing such that $\sum_{i=1}^n a_i < m$, hence $a_i \in \mathbb{Z}_m$, let $b := [b_1, \ldots, b_n] \in \mathbb{Z}_m^n$ such that $b_i = ca_i \in \mathbb{Z}/m\mathbb{Z}$. Since $c \in (\mathbb{Z}/m\mathbb{Z})^*$ the b_i are pairwise distinct, but b in general is not superincreasing, **disguising** the superincreasing sequence a.

Let $\mathcal{P} := \mathbb{Z}_2^n$ and $\mathcal{C} := \mathbb{Z}_{n(m-1)}$, and $\mathcal{K} := \{[c, a] \in \mathbb{Z}_m^* \times \mathbb{N}^n; a \text{ superincreasing}\}$. The public key is $b \in \mathbb{Z}_m^n$, the private key is $[m, c, a]$. Encryption is given as $E_b \colon \mathbb{Z}_2^n \to \mathbb{Z}_{n(m-1)} \colon [x_1, \ldots, x_n] \to \sum_{i=1}^n x_i b_i$. Decryption is given as follows: For $y \in \mathbb{Z}_{n(m-1)}$ let $z \in \mathbb{Z}_m$ such that $z = c^{-1}y \in \mathbb{Z}/m\mathbb{Z}$, then we let $D_{c,a} \colon \mathbb{Z}_{n(m-1)} \to \mathbb{Z}_2^n \colon y \mapsto S'(z; a)$; if the subset sum problem is not solvable we let $S'(z; a) := [0, \ldots, 0]$ instead. Then we have $E_b D_{c,a} = \mathrm{id}_{\mathcal{P}}$: For $[x_1, \ldots, x_n] \in \mathbb{Z}_2^n$ and $y := \sum_{i=1}^n x_i b_i \in \mathbb{Z}_{n(m-1)}$ we have $z = c^{-1}y = \sum_{i=1}^n x_i \cdot c^{-1} b_i = \sum_{i=1}^n x_i a_i \in \mathbb{Z}/m\mathbb{Z}$, implying $S'(z; a) = [x_1, \ldots, x_n]$.

The best known attack against the Merkle-Hellman cryptosystem is a ciphertext-only attack [Shamir, 1982; Lagarias, 1984] using **LLL lattice base reduction** [Lenstra-Lenstra-Lovász, 1982], leading to protocol failure, see [10, Ch.8.6.1].

(12.3) The Chor-Rivest cryptosystem [1988]. It is also based on the computational difficulty of the subset sum problem, but not on the idea of disguise:

a) For $n, k \in \mathbb{N}_0$ let $\binom{n}{k} := \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} \in \mathbb{N}_0$ be the associated binomial coefficient, where empty products are assumed to be equal to 1; we have $\binom{n}{0} = 1$, and $\binom{0}{k} = 0$ for $k \geq 1$, as well as $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ for $n, k \in \mathbb{N}$.

Given $n, k \in \mathbb{N}$ such that $k \leq n$, we consider the knapsack sequence $[\binom{m}{l} \in \mathbb{N}_0; m \in \{0, \ldots, n-1\}, l \in \{0, \ldots, \min\{k, m+1\}\}]$. Any $x \in \mathbb{Z}_{\binom{n}{k}}$ can be written as a sum $x = \sum_{i=1}^n x_i \cdot \binom{n-i}{k_i}$, where $x_i \in \{0, 1\}$ and $k_i \in \{0, \ldots, \min\{k, n-i+1\}\}$, such that precisely k of the x_i are equal to 1, and $[k_1, \ldots, k_n] = [k, \ldots, k, k-1, \ldots, k-1, \ldots, 0, \ldots, 0] \in \mathbb{N}_0^n$ where $k_{i+1} = k_i - 1$ if and only if $x_i = 1$:

We proceed by induction on $n \in \mathbb{N}$: If $n = 1$ then $k = 1$, hence $x = 0 = \binom{0}{1}$. If $n \geq 2$ then we let $k_1 := k$. If $0 \leq x < \binom{n-1}{k}$ then $k \leq n-1$, and letting $x_1 := 0$ we are done by induction. If $\binom{n-1}{k} \leq x < \binom{n}{k}$ then $0 \leq x - \binom{n-1}{k} < \binom{n}{k} - \binom{n-1}{k} = \binom{n-1}{k-1}$, and letting $x_1 := 1$ we are done by induction. ♯

Hence to recover x from $[x_1, \ldots, x_n] \in \{0, 1\}^n$, let $1 \leq i_1 < \cdots < i_k \leq n$ such that $x_i = 1$ if and only if $i \in \{i_1, \ldots, i_k\}$. Then we have $x = \sum_{j=1}^k \binom{n-i_j}{k-j+1}$.

b) Let $q \in \mathbb{N}$ be a prime power, let $2 \leq d \leq q$, and let $\mathcal{P} := \mathbb{Z}_{\binom{q}{d}}$ and $\mathcal{C} := \mathbb{Z}_{q^d - 1}$.

Fix a numbering $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$ of the field with q elements, let $\mathbb{F}_q \subseteq \mathbb{F}_{q^d}$ be the field with q^d elements, and let $\mathcal{K} := \{\rho \in \mathbb{F}_{q^d}^*; \rho \text{ primitive root}\}$. Given $\rho \in \mathcal{K}$, for $i \in \{1, \ldots, q\}$ let $a_i = \log_\rho(\rho+\alpha_i) \in \mathbb{Z}/(q^d-1)\mathbb{Z}$, i. e. we have $\rho+\alpha_i = \rho^{a_i} \in \mathbb{F}_{q^d}^*$. Then the private key is ρ, and the public key is $[q, d, a_1, \ldots, a_q]$.

The encryption function E_ρ is given as follows: For $x \in \mathcal{P}$ let $[x_1, \ldots, x_q] \in \{0,1\}^q$ such that precisely d of the x_i are equal to 1 as above. Then $y = E_\rho(x) \in \mathcal{C}$ is given by $y = \sum_{i=1}^q x_i a_i \in \mathbb{Z}/(q^d - 1)\mathbb{Z}$. Decryption is given as follows:

We have $\rho^y = \rho^{\sum_{i=1}^q x_i a_i} = \prod_{i=1}^q (\rho^{a_i})^{x_i} = \prod_{i=1}^q (\rho + \alpha_i)^{x_i} \in \mathbb{F}_{q^d}$. Let $\mu_\rho \in \mathbb{F}_q[X]$ be the minimum polynomial of ρ; hence μ_ρ is irreducible of degree d and $\mathbb{F}_q[X]/\mu_\rho\mathbb{F}_q[X] \cong \mathbb{F}_{q^d}$ as rings via $\overline{X} \mapsto \rho$. Let $f \in \mathbb{F}_q[X]$ such that $\deg(f) < d$ and $X^y \equiv f \pmod{\mu_\rho}$. Then $f + \mu_\rho \in \mathbb{F}_q[X]$ is monic of degree d such that $f + \mu_\rho \equiv X^y \pmod{\mu_\rho}$. Since $\prod_{i=1}^q (X + \alpha_i)^{x_i} \in \mathbb{F}_q[X]$ has the same properties we conclude $f + \mu_\rho = \prod_{i=1}^q (X + \alpha_i)^{x_i} \in \mathbb{F}_q[X]$. Hence $f + \mu_\rho$ splits into linear factors in $\mathbb{F}_q[X]$, and $[x_1, \ldots, x_q]$ can be recovered from them.

The factorisation of $f + \mu_\rho$ into linear factors can be computed by evaluating $f + \mu_\rho$ at $\alpha_i \in \mathbb{F}_q$, for $\in \{1, \ldots, q\}$. Computing the discrete logarithms a_i in $\mathbb{F}_{q^d}^*$ is feasible if $q^d - 1$ has only small prime factors. The recommended sizes are $q \sim 200$ and $d \sim 25$; one particular choice is $q = 197$, a prime, and $d = 24$, where $q^d - 1 \sim 10^{55}$ has largest prime factor $10\,316\,017 \sim 10^7$; we have $\binom{q}{d} \sim 4 \cdot 10^{30}$.

Drawbacks of the Chor-Rivest cryptosystem are message expansion, and its fairly large public keys. As it stands this cryptosystem is insecure, but against an only slightly more involved version there is no feasible attack known, provided parameters are chosen carefully to escape attacks against the underlying subset sum problem based on LLL lattice base reduction, see [10, Ch.8.6].

13 The Imai-Matsumoto cryptosystem

(13.1) The Imai-Matsumoto (IM) cryptosystem [1989]. Let $q \in \mathbb{N}$ be a 2-power, let $\mathbb{F}_q$ be the finite field with q elements. We consider a block cipher over $\mathbb{F}_q$ of block length $l \in \mathbb{N}$, i. e. we have $\mathcal{P} = \mathcal{C} = \mathbb{F}_q^l$. Encryption and decryption are given as follows:

Let $F := \mathbb{F}_{q^l}$ be the finite field with q^l elements, hence $\mathbb{F}_q \subseteq F$ is a field extension of degree $[F\colon \mathbb{F}_q] = l$, and let $\{\lambda_1, \ldots, \lambda_l\} \subseteq F$ be an $\mathbb{F}_q$-basis of F; hence we have a bijection $^-\colon \mathbb{F}_q^l \to F\colon x := [x_1, \ldots, x_l] \mapsto \sum_{i=1}^l x_i\lambda_i =: \overline{x}$, with inverse $\_\colon F \to \mathbb{F}_q^l$. Let $e := q^m + 1 \in \mathbb{Z}_{q^l-1}$, where $0 \neq m \in \mathbb{Z}_l$, such that $\gcd(e, q^l - 1) = 1$. Since $F^* \cong \mathbb{Z}/(q^l - 1)\mathbb{Z}$, the map $F^* \to F^*\colon \alpha \mapsto \alpha^e$ is bijective with inverse $F^* \to F^*\colon \alpha \mapsto \alpha^d$, where $d \in \mathbb{Z}_{q^l-1}$ such that $ed = 1 \in \mathbb{Z}/(q^l - 1)\mathbb{Z}$. Let $A = [a_{ij}]_{ij} \in \mathrm{GL}_l(\mathbb{F}_q)$ and $B = [b_{ij}]_{ij} \in \mathrm{GL}_l(\mathbb{F}_q)$, and $a = [a_1, \ldots, a_l] \in \mathbb{F}_q^l$ and $b = [b_1, \ldots, b_l] \in \mathbb{F}_q^l$, and let $\varphi_{A,a}\colon \mathbb{F}_q^l \to \mathbb{F}_q^l$ and $\varphi_{B,b}\colon \mathbb{F}_q^l \to \mathbb{F}_q^l$ be the associated affine $\mathbb{F}_q$-linear maps.

The private key is $[\lambda_1, \ldots, \lambda_l; \varphi_{A,a}, e, \varphi_{B,b}]$. Encryption and decryption are given by $\mathbb{F}_q^l \xrightarrow{\varphi_{A,a}} \mathbb{F}_q^l \xrightarrow{\overline{}} F \xrightarrow{\alpha \mapsto \alpha^e} F \xrightarrow{\overline{}} \mathbb{F}_q^l \xrightarrow{\varphi_{B,b}} \mathbb{F}_q^l$, and $\mathbb{F}_q^l \xrightarrow{\varphi_{B,b}^{-1}} \mathbb{F}_q^l \xrightarrow{\overline{}} F \xrightarrow{\alpha \mapsto \alpha^d} F \xrightarrow{\overline{}} \mathbb{F}_q^l \xrightarrow{\varphi_{A,a}^{-1}} \mathbb{F}_q^l$, respectively; thus encryption is given by powering in F^* which is disguised by the affine $\mathbb{F}_q$-linear maps.

The public key is given as follows: The **Frobenius map** $F \to F\colon \alpha \mapsto \alpha^{q^m}$ is a field automorphism fixing $\mathbb{F}_q$ elementwise, hence is a bijective $\mathbb{F}_q$-linear map. Let $Q = [q_{ij}]_{ij} \in \mathrm{GL}_l(\mathbb{F}_q)$ be its matrix with respect to the chosen $\mathbb{F}_q$-basis of F. For $i, j \in \{1, \ldots, l\}$ we have $\lambda_i \lambda_j = \sum_{k=1}^l p_{ijk} \lambda_k$, where $p_{ijk} \in \mathbb{F}_q$, and we let $P_i := [p_{ijk}]_{jk} \in \mathrm{GL}_l(\mathbb{F}_q)$.

Let $x = [x_1, \ldots, x_l] \in \mathbb{F}_q^l$ be a plaintext with ciphertext $y = [y_1, \ldots, y_l] \in \mathbb{F}_q^l$. Let $u = [u_1, \ldots, u_l] := \varphi_{A,a}(x) \in \mathbb{F}_q^l$ and $v = [v_1, \ldots, v_l] := \varphi_{B,b}^{-1}(y) \in \mathbb{F}_q^l$, hence we have $u_i = \sum_{s=1}^l x_s a_{si} + a_i$ and $y_t = \sum_{j=1}^l v_j b_{jt} + b_t$ for $i, t \in \{1, \ldots, l\}$. Using this we compute $\overline{v} = \overline{u}^e = \overline{u}^{q^m} \cdot \overline{u} = (\sum_{i=1}^l u_i \lambda_i^{q^m}) \cdot (\sum_{j=1}^l u_j \lambda_j) = (\sum_{i=1}^l \sum_{k=1}^l u_i q_{ik} \lambda_k) \cdot (\sum_{j=1}^l u_j \lambda_j) = \sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l u_i u_j q_{ik} \lambda_k \lambda_j$, and hence we obtain $\overline{v} = \sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l \sum_{r=1}^l u_i u_j q_{ik} p_{kjr} \lambda_r$, which finally implies $v_r = \sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l u_i u_j q_{ik} p_{kjr}$ for $r \in \{1, \ldots, l\}$. From this we obtain $y_t = b_t + \sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l \sum_{r=1}^l (\sum_{s=1}^l x_s a_{si} + a_i)(\sum_{s=1}^l x_s a_{sj} + a_j) q_{ik} p_{kjr} b_{rt}$. The public key consists of the above quadratic equations describing the entries of y in terms of the entries of x. Using the private key we find quadratic equations to decrypt y again, while an adversary has to solve a system of nonlinear equations for the entries of x.

(13.2) Cryptanalysis of the IM cryptosystem [Patarin, 1995]. The cryptanalysis is based on the following observation: From $\overline{v} = \overline{u}^{q^m+1}$ we deduce $\overline{v}^{q^m-1} = \overline{u}^{(q^m+1)(q^m-1)} = \overline{u}^{q^{2m}-1}$, and thus necessarily $\overline{u}^{q^{2m}} \cdot \overline{v} = \overline{u} \cdot \overline{v}^{q^m}$.

If $0 \neq v \in \mathbb{F}_q^l$ is fixed, the set of $0 \neq u' \in \mathbb{F}_q^l$ fulfilling $\overline{u'}^{q^{2m}} \cdot \overline{v} = \overline{u'} \cdot \overline{v}^{q^m}$ is given as follows: Let $u \in \mathbb{F}_q^l$ be the unique element such that $\overline{u}^{q^m+1} = \overline{v}$, thus $\overline{v}^{q^m-1} = \overline{u}^{q^{2m}-1}$. If $\overline{u'}^{q^{2m}} \cdot \overline{v} = \overline{u'} \cdot \overline{v}^{q^m}$ for some $u' \neq 0$, then we also have $\overline{v}^{q^m-1} = \overline{u'}^{q^{2m}-1}$. Thus we have $\overline{u}^{(q^m-1)(q^m+1)} = \overline{u'}^{(q^m-1)(q^m+1)}$, and since $\gcd(q^m+1, q^n-1) = 1$ we conclude $\overline{u}^{q^m-1} = \overline{u'}^{q^m-1}$, and hence $\overline{u'} \cdot \overline{u}^{-1} \in F$ is a (q^m-1)-st root of unity. Conversely, if $\zeta \in F$ is a (q^m-1)-st root of unity, then from $\overline{u}^{q^{2m}} \cdot \overline{v} = \overline{u} \cdot \overline{v}^{q^m}$ we get $(\overline{u} \cdot \zeta)^{q^{2m}} \cdot \overline{v} = (\overline{u} \cdot \zeta) \cdot \overline{v}^{q^m}$.

From the Euclidean algorithm we get $\gcd(q^m-1, q^l-1) = q^g - 1$ where $g := \gcd(m, l) \in \mathbb{N}$, hence $\{\zeta \in F^*; \zeta^{q^m-1} = 1\} = \{\zeta \in F^*; \zeta^{q^g-1} = 1\} = K^*$, where $\mathbb{F}_q \subseteq K \subseteq F$ is the unique subfield such that $[K\colon \mathbb{F}_q] = g$. Hence for $v \neq 0$ the above set of solutions, including $u' = 0$, is given as $\{u' \in \mathbb{F}_q^l; \overline{u'} \in \overline{u} \cdot K\}$, which is a g-dimensional $\mathbb{F}_q$-subspace of $\mathbb{F}_q^l$. Transforming to plaintext-ciphertext pairs using $\varphi_{A,a}^{-1}$ and $\varphi_{B,b}$ shows that, given a ciphertext $y \in \mathbb{F}_q^l$ such that $v \neq 0$, the associated plaintext $x \in \mathbb{F}_q^l$ is contained in a g-dimensional affine $\mathbb{F}_q$-subspace of $\mathbb{F}_q^l$, thus having cardinality q^g.

Hence if g is small, an exhaustive search attack is feasible; from $0 \neq m \in \mathbb{Z}_l$ we get $g = \gcd(m, l) < l$ anyway. Assume we have $g = \frac{l}{2}$, hence $2m = l$ and $\gcd(q^m + 1, q^n - 1) = \gcd(q^m + 1, q^{2m} - 1) = q^m + 1 \neq 1$, a contradiction. Hence we have $g \leq \frac{l}{3}$, where this case can indeed occur: If $3m = l$, then since $\gcd(q^m - 1, q^{3m} - 1) = q^m - 1 = \gcd(q^{2m} - 1, q^{3m} - 1)$ we conclude $\gcd(q^m + 1, q^{3m} - 1) = 1$. Thus to prevent this attack, l has to be chosen large, hence the cryptosystem is either insecure or inefficient. It remains to show how the above affine solution space can be found:

From $\overline{u}^{q^{2m}} \cdot \overline{v} = (\overline{u}^{q^m})^{q^m} = \overline{u} \cdot \overline{v}^{q^m}$, for all $v \in \mathbb{F}_q^l$, we obtain $(\sum_{i=1}^l u_i (\lambda_i^{q^m})^{q^m}) \cdot (\sum_{j=1}^l v_j \lambda_j) = (\sum_{i=1}^l u_i \lambda_i) \cdot (\sum_{j=1}^l v_j \lambda_j^{q^m})$, hence we get $(\sum_{i=1}^l \sum_{k=1}^l u_i q_{ik} \lambda_k^{q^m}) \cdot (\sum_{j=1}^l v_j \lambda_j) = (\sum_{i=1}^l \sum_{k=1}^l \sum_{h=1}^l u_i q_{ik} q_{kh} \lambda_h) \cdot (\sum_{j=1}^l v_j \lambda_j) = (\sum_{i=1}^l u_i \lambda_i) \cdot (\sum_{j=1}^l \sum_{k=1}^l v_j q_{jk} \lambda_k)$, thus we obtain $\sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l \sum_{h=1}^l u_i v_j q_{ik} q_{kh} p_{hjr} = \sum_{i=1}^l \sum_{j=1}^l \sum_{k=1}^l u_i v_j q_{jk} p_{ikr}$ for $r \in \{1, \ldots, l\}$. Using $u_i = \sum_{s=1}^l x_s a_{si} + a_i$ and $v_j = \sum_{t=1}^l (y_t - b_t) b'_{tj}$, where $B^{-1} = [b'_{ij}]_{ij} \in \mathrm{GL}_l(\mathbb{F}_q)$, leads to equations $(\sum_{s=1}^l \sum_{t=1}^l \alpha_{str} x_s y_t) + (\sum_{s=1}^l \beta_{sr} x_s) + (\sum_{t=1}^l \gamma_{tr} y_t) + \delta_r = 0$ for $r \in \{1, \ldots, l\}$ where $\alpha_{str}, \beta_{sr}, \gamma_{tr}, \delta_r \in \mathbb{F}_q$.

The latter equations are linear in x, hence once the coefficients are known finding the candidate x for a given y amounts to solving a system of linear equations; since the affine solution space is g-dimensional, it is described by $l - g$ linearly independent equations for x. We have l disjoint identical systems of equations $(\sum_{s=1}^l \sum_{t=1}^l \alpha_{st} x_s y_t) + (\sum_{s=1}^l \beta_s x_s) + (\sum_{t=1}^l \gamma_t y_t) + \delta = 0$. Since each plaintext-ciphertext pair yields a linear equation for the $l^2 + 2l + 1 = (l+1)^2$ coefficients $\alpha_{st}, \beta_s, \gamma_t, \delta$, the latter can be found by a chosen-plaintext attack. Since there are $(l+1)^2$ coefficients, $(l+1)^2 - (l-g)$ linearly independent equations coming from plaintext-ciphertext pairs are needed to give $l - g$ linearly independent equations for x.

There is a total of q^l plaintext-ciphertext pairs, whose number thus grows exponentially with q. Hence if q is large, a negligible fraction of all plaintext-ciphertext pairs suffices to find a maximal linearly independent set of linear equations. If q is small there might be less than $(l+1)^2 - (l-g)$ linearly independent equations coming from plaintext-ciphertext pairs, implying that for a given ciphertext y there is an affine solution space of $\mathbb{F}_q$-dimension less than g, being described by more than $l - g$ linearly independent equations for x.

(13.3) Example. Let $q := 2$ and $l := 3$. Since $T^3 + T + 1 \in \mathbb{F}_2[T]$ is irreducible we have $\mathbb{F}_8 \cong F := \mathbb{F}_2[T]/(T^3 + T + 1)\mathbb{F}_2[T]$. Let $\{1, T, T^2\} \subseteq F$ be the chosen $\mathbb{F}_2$-basis. Let $m := 2$, hence we have $e = q^m + 1 = 5$ and $\gcd(q^l - 1, e) = \gcd(7, 5) = 1$, and thus $d = 3$ and $g = \gcd(m, l) = 1$. Let $a := [1, 0, 1] \in \mathbb{F}_2^3$ and $b := [1, 0, 0] \in \mathbb{F}_2^3$, and

$$A = \begin{bmatrix} . & 1 & . \\ 1 & 1 & . \\ . & 1 & 1 \end{bmatrix} \in \mathrm{GL}_3(\mathbb{F}_2) \quad \text{and} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ . & 1 & 1 \\ . & . & 1 \end{bmatrix} \in \mathrm{GL}_3(\mathbb{F}_2).$$

By computing $(T^i)^4 \in F$ for $i \in \{0, \ldots, 2\}$ we obtain $Q = \begin{bmatrix} 1 & . & . \\ . & 1 & 1 \\ . & 1 & . \end{bmatrix} \in$ $\mathrm{GL}_3(\mathbb{F}_2)$, and by computing the products $T^iT^j \in F$ for $i, j \in \{0, \ldots, 2\}$ we obtain $P_1, P_2, P_3 \in \mathrm{GL}_3(\mathbb{F}_2)$ where

$$P_1 = \begin{bmatrix} 1 & . & . \\ . & 1 & . \\ . & . & 1 \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} . & 1 & . \\ . & . & 1 \\ 1 & 1 & . \end{bmatrix} \quad \text{and} \quad P_3 = \begin{bmatrix} . & . & 1 \\ 1 & 1 & . \\ . & 1 & 1 \end{bmatrix}.$$

For $u = [u_1, u_2, u_3] \in \mathbb{F}_2^3$ and $v = [v_1, v_2, v_3] \in \mathbb{F}_2^3$ we obtain $\overline{u}^e = \overline{u}^4 \cdot \overline{u} = (u_1 + (u_2+u_3)T+u_2T^2)\cdot(u_1+u_2T+u_3T^2) \in F$. From this we obtain $v_1+v_2T+v_3T^2 = \overline{v} = \overline{u}^e = (u_1^2+u_2^2+u_2u_3+u_3^2)+(u_1u_3+u_2^2+u_3^2)T+(u_1u_2+u_1u_3+u_2^2)T^2 \in F$, thus $v_1 = u_1^2+u_2^2+u_2u_3+u_3^2$ and $v_2 = u_1u_3+u_2^2+u_3^2$ and $v_3 = u_1u_2+u_1u_3+u_2^2$.

For $x = [x_1, x_2, x_3] \in \mathbb{F}_2^3$ and $y = [y_1, y_2, y_3] \in \mathbb{F}_2^3$ we have $u_1 = x_2 + 1$ and $u_2 = x_1 + x_2 + x_3$ and $u_3 = x_3 + 1$, as well as $y_1 = v_1 + 1$ and $y_2 = v_1 + v_2$ and $y_3 = v_1 + v_2 + v_3$. This yields $v_1 = (x_2 + 1)^2 + (x_1 + x_2 + x_3)^2 + (x_1 + x_2 + x_3)(x_3 + 1) + (x_3 + 1)^2 = x_1^2 + x_1x_3 + x_2x_3 + x_3^2 + x_1 + x_2 + x_3$ and $v_2 = (x_2+1)(x_3+1)+(x_1+x_2+x_3)^2+(x_3+1)^2 = x_1^2+x_2^2+x_2x_3+x_2+x_3$ and $v_3 = (x_2+1)(x_1+x_2+x_3)+(x_2+1)(x_3+1)+(x_1+x_2+x_3)^2 = x_1^2+x_1x_2+x_3^2+x_1+1$. Hence the public key is

$$\begin{array}{rcl} y_1 & = & x_1^2 + x_1x_3 + x_2x_3 + x_3^2 + x_1 + x_2 + x_3 + 1, \\ y_2 & = & x_1x_3 + x_2^2 + x_3^2 + x_1, \\ y_3 & = & x_1^2 + x_1x_2 + x_1x_3 + x_2^2 + 1. \end{array}$$

Hence the plaintext-ciphertext pairs $[x; y] = [x_1, x_2, x_3; y_1, y_2, y_3] \in \mathbb{F}_2^3 \times \mathbb{F}_2^3$ are

$$\left[\begin{array}{ccc|ccc} . & . & . & 1 & . & 1 \\ . & . & 1 & 1 & 1 & 1 \\ . & 1 & . & . & 1 & . \\ . & 1 & 1 & 1 & . & . \\ 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & 1 \\ 1 & 1 & . & . & . & . \\ 1 & 1 & 1 & . & . & 1 \end{array}\right].$$

Running through plaintext-ciphertext pairs yields the following relation matrix $M \in \mathbb{F}_2^{8\times 16}$, i. e. $M \cdot [\alpha_{11}, \alpha_{12}, \ldots, \alpha_{33}, \beta_1, \ldots, \beta_3, \gamma_1, \ldots, \gamma_3, \delta]^{\mathrm{tr}} = 0$, i. e.

$[\alpha_{11}, \alpha_{12}, \ldots, \alpha_{33}, \beta_1, \ldots, \beta_3, \gamma_1, \ldots, \gamma_3, \delta] \in \ker(M^{\mathrm{tr}}) \leq \mathbb{F}_2^{16}$:

$$M = \left[\begin{array}{ccccccccc|ccc|ccc|c}
. & . & . & . & . & . & . & . & . & . & . & . & 1 & . & 1 & 1 \\
. & . & . & . & . & . & 1 & 1 & 1 & . & . & 1 & 1 & 1 & 1 & 1 \\
. & . & . & . & 1 & . & . & . & . & . & 1 & . & . & 1 & . & 1 \\
. & . & . & 1 & . & . & 1 & . & . & . & 1 & 1 & 1 & . & . & 1 \\
1 & 1 & . & . & . & . & . & . & . & 1 & . & . & 1 & 1 & . & 1 \\
. & 1 & 1 & . & . & . & . & 1 & 1 & 1 & . & 1 & . & 1 & 1 & 1 \\
. & . & . & . & . & . & . & . & . & 1 & 1 & . & . & . & . & 1 \\
. & . & 1 & . & . & 1 & . & . & 1 & 1 & 1 & 1 & . & . & 1 & 1
\end{array}\right]$$

It turns out that M has maximal rank $\mathrm{rk}_{\mathbb{F}_2}(M) = 8$, and an $\mathbb{F}_2$-basis of $\ker(M^{\mathrm{tr}})$ is given as follows:

$$\left[\begin{array}{ccccccccc|ccc|ccc|c}
1 & 1 & 1 & . & . & 1 & . & . & . & . & . & . & . & . & . & . \\
1 & 1 & . & 1 & . & . & 1 & 1 & . & . & . & . & . & . & . & . \\
. & . & 1 & 1 & . & . & 1 & . & 1 & . & . & . & . & . & . & . \\
. & 1 & . & 1 & 1 & . & . & . & . & 1 & 1 & . & . & . & . & . \\
. & . & 1 & . & . & . & 1 & . & . & . & . & 1 & . & . & . & . \\
. & 1 & . & 1 & 1 & . & 1 & . & . & . & . & . & . & 1 & . & . \\
1 & . & 1 & 1 & . & . & . & . & . & . & . & . & 1 & . & 1 & . \\
1 & . & . & . & 1 & . & . & . & . & 1 & . & . & 1 & . & . & 1
\end{array}\right]$$

Each of the 3-subsets $\{1, 4, 7\}$, $\{2, 5, 8\}$, $\{3, 6, 9\}$ and $\{10, 11, 12\}$ of columns is $\mathbb{F}_2$-linearly independent, thus for all ciphertexts $y \in \mathbb{F}_2^3$ the matrix of the resulting system of linear equations for the associated plaintext $x \in \mathbb{F}_2^3$ has rank 3, and since there is a solution, the affine solution space for x is 0-dimensional, hence x is uniquely determined. The cryptanalysis only ensures 1-dimensional affine solution spaces, but here the plaintext-ciphertext pairs do actually fulfil more equations than the generic ones:

The input data is also valid for the case $q := 4$ and $m := 1$, where since $T^3 + T + 1 \in \mathbb{F}_4[T]$ still is irreducible we let $\mathbb{F}_{64} \cong F' := \mathbb{F}_4[T]/(T^3 + T + 1)\mathbb{F}_4[T]$. We still have $e = q^m + 1 = 5$ and $\gcd(e, q^l - 1) = \gcd(5, 63) = 1$ and $g = \gcd(m, l) = 1$. Running through all $4^3 = 64$ plaintext-ciphertext pairs yields a matrix $M' \in \mathbb{F}_4^{64 \times 16}$ such that $\mathrm{rk}_{\mathbb{F}_4}(M') = 14$, and we get the following $\mathbb{F}_4$-basis of $\ker(M'^{\mathrm{tr}}) \leq \mathbb{F}_4^{64}$ consisting of $l - g = 2$ elements; we have $\ker(M'^{\mathrm{tr}}) \leq (\ker(M^{\mathrm{tr}}) \otimes_{\mathbb{F}_2} \mathbb{F}_4)$:

$$\left[\begin{array}{ccccccccc|ccc|ccc|c}
1 & 1 & 1 & 1 & . & . & 1 & 1 & . & 1 & 1 & 1 & . & 1 & . & . \\
1 & . & 1 & . & 1 & 1 & 1 & 1 & 1 & 1 & . & 1 & 1 & . & . & 1
\end{array}\right]$$

For generalisations of Imai-Matsumoto cryptosystem and their cryptanalysis, see [8, Ch.4.2, 4.3].

III Integer arithmetic

14 Computational complexity

(14.1) Turing machines. The standard model of algorithmic computing is performing operations on finite strings of letters out of a finite alphabet being written onto an infinite tape, using a machine running back and forth on the tape reading and writing letters according to specified rules. By **Church's Hypothesis** this idea precisely covers the intuitive notion of algorithmic computability. An early occurrence of this type of question is **Hilbert's 10th problem** on the decidability of the solubility of Diophantine equations; it was solved to the negative by Matijasevich [1972].

A **(deterministic) Turing machine** over an alphabet $\mathcal{X}$ is a triple $\mathcal{T} := [\mathcal{X} \,\dot\cup\, \mathcal{Y}, \mathcal{S}, \tau]$, where $\mathcal{Y}$ is a **working** alphabet, in particular containing a **blank** letter $\_ \in \mathcal{Y}$, an **accepting** letter $\underline{1}$, and a **rejecting** or **failure** letter $\underline{0}$, where $\mathcal{S}$ is a finite set of **states**, in particular containing an **initial** state s_0 and a **halting** state s_∞, and where τ is a **transition function**

$$\tau\colon (\mathcal{X} \,\dot\cup\, \mathcal{Y}) \times (\mathcal{S} \setminus \{s_\infty\}) \longrightarrow (\mathcal{X} \,\dot\cup\, \mathcal{Y}) \times \{\leftarrow, \uparrow, \rightarrow\} \times \mathcal{S}.$$

$\mathcal{T}$ acts on the set $(\mathcal{X} \,\dot\cup\, \mathcal{Y})^* \times \mathcal{S} \times (\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$ of **configurations** as follows: The **initial configurations** are given as $[\_, s_0, u]$, where $u \in \mathcal{X}^*$ is called an **input**; an input of several $u_1, \ldots, u_n \in \mathcal{X}^*$ is encoded as $u_1\_u_2\_\ldots\_u_n \in (\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$.

Let $[v, s, w]$ be a configuration, where $s \in \mathcal{S} \setminus \{s_\infty\}$. If $\epsilon \neq v, w \in (\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$, let $v = v'x$ and $w = yw'$, where $x, y \in \mathcal{X} \,\dot\cup\, \mathcal{Y}$; if $v = \epsilon$, let $v' := \epsilon$ and $x := \_$; if $w = \epsilon$, let $w' := \epsilon$ and $y := \_$. Then $\mathcal{T}$ induces the **transition**

$$[v, s, w] \mapsto \begin{cases} [v, \quad s', \quad zw'], & \text{if } \tau(y, s) = [z, \uparrow, s'], \\ [vz, \quad s', \quad w'], & \text{if } \tau(y, s) = [z, \rightarrow, s'], \\ [v', \quad s', \quad xzw'], & \text{if } \tau(y, s) = [z, \leftarrow, s']. \end{cases}$$

For a configuration $[v, s_\infty, w]$ no transition is defined and $\mathcal{T}$ **halts**. We assume that for all inputs leading to such a halting configuration we are in one and the same of the following cases, depending on whether we consider a **decision problem** or a **function problem**: Either we have $w \in \underline{1}(\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$ or $w \in \underline{0}(\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$, i. e. $\mathcal{T}$ **accepts** or **rejects**, respectively; or we have $w \in \underline{0}(\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$ or $w \in w'\_(\mathcal{X} \,\dot\cup\, \mathcal{Y})^*$, where $w' \in \mathcal{X}^*$, i. e. $\mathcal{T}$ **fails** or **outputs** w'.

(14.2) Example. Let $\mathcal{X} := \{1\}$ and $\mathcal{S} := \{s_0, s_1, s_\infty\}$, and let $\mathcal{T}$ be given by

τ	$\_$	1
s_0	$[1, \leftarrow, s_1]$	$[1, \rightarrow, s_0]$
s_1	$[\_, \rightarrow, s_\infty]$	$[1, \leftarrow, s_1]$

.

Hence upon input $11 \in \mathcal{X}^2$ we obtain

$$\_\_\boxed{s_0}11\_\_ \quad\mapsto\quad \_\_1\boxed{s_0}1\_\_ \quad\mapsto\quad \_\_11\boxed{s_0}\_\_ \quad\mapsto$$

$$\_\_1\boxed{s_1}11\_ \quad\mapsto\quad \_\_\boxed{s_1}111\_ \quad\mapsto\quad \_\boxed{s_1}\_111\_ \quad\mapsto\quad \_\_\boxed{s_\infty}111\_,$$

and for $\epsilon \in \mathcal{X}^0$ we obtain $\_\_\boxed{s_0}\_\_ \quad\mapsto\quad \_\boxed{s_1}\_1\_ \quad\mapsto\quad \_\_\boxed{s_\infty}1\_ \quad .$

$\mathcal{T}$ computes the successor function $\mathbb{N}_0 \to \mathbb{N}\colon n \mapsto n+1$, where $\mathbb{N}_0$ is given in **unary encoding**.

(14.3) Decision problems. Let $\mathcal{X}$ be an alphabet, and let $\mathcal{L} \subseteq \mathcal{X}^*$ be a language. Then $\mathcal{L}$ is called **decidable (recursive)**, if there is a Turing machine $\mathcal{T}$ **deciding** $\mathcal{L}$, i. e. $\mathcal{T}$ halts for all $w \in \mathcal{X}^*$, and accepts w if and only if $w \in \mathcal{L}$, otherwise rejects w; and $\mathcal{L}$ is called **recursively enumerable**, if there is a Turing machine $\mathcal{T}$ **accepting** $\mathcal{L}$, i. e. $\mathcal{T}$ halts for $w \in \mathcal{X}^*$ if and only if $w \in \mathcal{L}$. Hence if $\mathcal{L}$ is decidable, then it is recursively enumerable: Let $\mathcal{T}$ decide $\mathcal{L}$, then $\mathcal{T}'$ accepting $\mathcal{L}$ is a copy of $\mathcal{T}$, except that whenever $\mathcal{T}$ rejects an input, then $\mathcal{T}'$ enters an infinite loop.

A Turing machine $\mathcal{T}$ deciding a language $\mathcal{L} \subseteq \mathcal{X}^*$ is called to run in **time** $f\colon \{n \in \mathbb{N}; n \geq N\} \to \mathbb{R}_{>0}$, where $N \in \mathbb{N}$, if $\mathcal{T}$ halts after at most $f(l(w))$ transitions, for all $w \in \mathcal{X}^*$ such that $l(w) \geq N$. The **complexity class** $\mathsf{TIME}(f) \subseteq \mathrm{Pot}(\mathcal{X}^*)$ is the set of all languages being decidable in time f. In particular, we have the complexity classes $\mathsf{P} := \bigcup_{k \in \mathbb{N}} \mathsf{TIME}(n^k)$ and $\mathsf{EXP} := \bigcup_{k \in \mathbb{N}} \mathsf{TIME}(c^{n^k})$, where $c > 1$, of languages being decidable in **polynomial** and **exponential** time, respectively; EXP does not depend on the choice of $c > 1$. We have $\mathsf{P} = \mathsf{coP}$, where coP is the complexity class of languages $\mathcal{L} \subseteq \mathcal{X}^*$ such that $(\mathcal{X}^* \setminus \mathcal{L}) \in \mathsf{P}$.

(14.4) Non-deterministic Turing machines. A **non-deterministic Turing machine** over an alphabet $\mathcal{X}$ is a triple $\mathcal{T} := [\mathcal{X} \,\dot{\cup}\, \mathcal{Y}, \mathcal{S}, \tau]$, where $\mathcal{X} \,\dot{\cup}\, \mathcal{Y}$ and $\mathcal{S}$ are as in (14.1), while the transition function

$$\tau\colon (\mathcal{X} \,\dot{\cup}\, \mathcal{Y}) \times (\mathcal{S} \setminus \{s_\infty\}) \longrightarrow \mathrm{Pot}((\mathcal{X} \,\dot{\cup}\, \mathcal{Y}) \times \{\leftarrow, \uparrow, \rightarrow\} \times \mathcal{S})$$

allows for **choices** and thus **branching**. Let the **non-determinateness** be defined as $d_{\mathcal{T}} := \max\{|\tau(x,s)|; x \in \mathcal{X} \,\dot{\cup}\, \mathcal{Y}, s \in \mathcal{S} \setminus \{s_\infty\}\} \in \mathbb{N}$. The machine $\mathcal{T}$ halts if no further transition in either branch is possible. We assume that for all inputs $\mathcal{T}$ on halting either accepts or rejects, or outputs; for acceptance, rejection or output one of the branches is chosen randomly.

A language $\mathcal{L} \subseteq \mathcal{X}^*$ is called **non-deterministically decidable**, if there is a non-deterministic Turing machine $\mathcal{T}$ **deciding** $\mathcal{L}$, i. e. $\mathcal{T}$ halts for all $w \in \mathcal{X}^*$, and we have $w \in \mathcal{L}$ if and only if there is a branch accepting w, otherwise all branches reject w; acceptance and rejection are treated asymmetrically.

The complexity class $\mathsf{NTIME}(f)$ is the set of all languages being non-deterministically decidable in **time** f. In particular, we have the complexity class $\mathsf{NP} :=$

$\bigcup_{k\in\mathbb{N}} \mathsf{NTIME}(n^k)$ of languages being decidable in **non-deterministic polynomial** time. Let coNP be the complexity class of languages $\mathcal{L} \subseteq \mathcal{X}^*$ such that $(\mathcal{X}^* \setminus \mathcal{L}) \in \mathsf{NP}$; we have $\mathsf{P} \subseteq \mathsf{NP} \cap \mathsf{coNP}$. It is conjectured that $\mathsf{P} \neq \mathsf{NP}$ and $\mathsf{NP} \neq \mathsf{coNP}$ and $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$, the most outstanding open problem of computational complexity theory.

(14.5) Theorem. We have $\mathsf{NTIME}(f) \subseteq \bigcup_{c>1} \mathsf{TIME}(c^f)$, thus in particular we have $\mathsf{NP} \subseteq \mathsf{EXP}$.

Proof. Let $\mathcal{L}$ be non-deterministically decidable by a Turing machine $\mathcal{T}$, having non-determinateness $d_\mathcal{T} \geq 2$ and running in time f. Each finite sequence of choices of $\mathcal{T}$ can be encoded $d_\mathcal{T}$-adically as $\sum_{i\geq 1} c_i d_\mathcal{T}^{i-1} \in \mathbb{N}_0$, where $c_i \in \mathbb{Z}_d$.

Let $\mathcal{T}'$ be a 3-string Turing machine, where the input is kept on string 1, while on string 2 a counter enumerates $\mathbb{N}_0$ in $d_\mathcal{T}$-adic representation, thus producing all possible sequences of choices of increasing lengths. On string 3 the computation of $\mathcal{T}$, with the sequence of choices determined by string 2, is done. If for some finite sequence of choices $\mathcal{T}$ accepts, then so does $\mathcal{T}'$. Otherwise $\mathcal{T}'$ runs until some $t \in \mathbb{N}_0$ is reached such that for no sequence of choices of length t a sequence of choices of length $t+1$ is possible, in this case $\mathcal{T}'$ rejects.

The running time of $\mathcal{T}'$ is the time needed to do the computation for a fixed sequence, times the number of sequences to be considered. Letting n be the input length, the former is in $O(f(n))$, and the latter is in $O(\sum_{t=1}^{f(n)} (d_\mathcal{T})^t) = O((d_\mathcal{T})^{f(n)+1})$. Then $\mathcal{T}'$ can be simulated by a conventional Turing machine at the expense of squaring the running time. ♯

(14.6) Theorem. Let $\mathcal{X}$ be an alphabet such that $|\mathcal{X}| \geq 2$, and let $\mathcal{L} \subseteq \mathcal{X}^*$ be a language. Then we have $\mathcal{L} \in \mathsf{NP}$ if and only if there is a relation $\mathcal{R} \subseteq \mathcal{X}^* \times \mathcal{X}^*$ having the following properties:
i) We have $\mathcal{L} = \{w \in \mathcal{X}^*; [w,v] \in \mathcal{R} \text{ for some } v \in \mathcal{X}^*\}$.
ii) There is $k \in \mathbb{N}$ such that $l(v) \leq l(w)^k$ for all $[w,v] \in \mathcal{R}$.
iii) Letting $\mathcal{L}_\mathcal{R} := \{w\_v; [w,v] \in \mathcal{R}\} \subseteq \mathcal{X}^*\_\mathcal{X}^*$, we have $\mathcal{L}_\mathcal{R} \in \mathsf{P}$.

Given $w \in \mathcal{L}$, an element $v \in \mathcal{X}^*$ such that $[w,v] \in \mathcal{R}$ is called a **polynomial certificate** for w.

Proof. Let $\mathcal{R}$ fulfil the above conditions. Then $\mathcal{L}$ is decided by a non-deterministic Turing machine, which for $w \in \mathcal{X}^*$ first finds a certificate $v \in \mathcal{X}^*$ of polynomial length $l(v) \leq l(w)^k$, hence in polynomial time, and then decides in polynomial time whether $[w,v] \in \mathcal{R}$. Hence we have $\mathcal{L} \in \mathsf{NP}$.

Conversely, let $\mathcal{L} \in \mathsf{NP}$ be decided by a non-deterministic Turing machine $\mathcal{T}$, running in polynomial time and having non-determinateness $d_\mathcal{T}$. Each finite sequence of choices of $\mathcal{T}$ can be encoded $d_\mathcal{T}$-adically into an element of $\mathbb{N}_0$, and hence $|\mathcal{X}|$-adically into an element of $\mathcal{X}^*$. Thus we define $\mathcal{R} \subseteq \mathcal{X}^* \times \mathcal{X}^*$ by letting $[w,v] \in \mathcal{R}$ if and only if $v \in \mathcal{X}^*$ is the encoding of a sequence of choices of an accepting computation for $w \in \mathcal{X}^*$. Hence by construction of $\mathcal{R}$ we have (i) and (ii). Moreover, for w_v it can be checked in linear time whether v indeed encodes an accepting computation for w, hence we also have (iii). $\sharp$

(14.7) Randomised machines. a) A **(one-sided) Monte-Carlo machine** for a language $\mathcal{L} \subseteq \mathcal{X}^*$ is a non-deterministic Turing machine $\mathcal{T}$ halting for all $w \in \mathcal{X}^*$, having an **error bound** $0 < \epsilon < 1$ such that $\mathcal{T}$ accepts $w \in \mathcal{L}$ in at least a fraction of ϵ of the branches, while $\mathcal{T}$ rejects $w \notin \mathcal{L}$ in all branches. Hence acceptance is correct, but rejection might be incorrect with an error probability $1 - \epsilon$; we may fix an error bound $0 < \epsilon_0 < 1$ a priorly: If $\epsilon < \epsilon_0$, then $\mathcal{T}$ is repeated k times, until $(1 - \epsilon)^k \leq (1 - \epsilon_0)$.

The complexity class RP of languages being decidable in **randomised polynomial** time is the set of languages possessing a Monte-Carlo machine running in polynomial time. Hence we have $\mathsf{P} \subseteq \mathsf{RP} \subseteq \mathsf{NP}$. Let coRP be the complexity class of languages $\mathcal{L} \subseteq \mathcal{X}^*$ such that $(\mathcal{X}^* \setminus \mathcal{L}) \in \mathsf{RP}$, and let $\mathsf{ZPP} := \mathsf{RP} \cap \mathsf{coRP}$ be the complexity class of languages being decidable in randomised polynomial time with **zero probability error**:

For $\mathcal{L} \in \mathsf{ZPP}$ let $\mathcal{T}'$ and $\mathcal{T}''$ be Monte-Carlo machines for $\mathcal{L}$ and $\mathcal{X}^* \setminus \mathcal{L}$, respectively, both with error bound $0 < \epsilon < 1$. A **Las-Vegas machine** for $\mathcal{L}$ is a non-deterministic Turing machine $\mathcal{T}$ defined as follows: $\mathcal{T}$ runs both $\mathcal{T}'$ and $\mathcal{T}''$, if $\mathcal{T}'$ accepts then $\mathcal{T}$ accepts, if $\mathcal{T}''$ accepts then $\mathcal{T}$ rejects, and otherwise repeats. Hence $\mathcal{T}$ not necessarily halts, but if it does then the result is correct; $\mathcal{T}$ halts after at most k repetitions with a probability of at least $1 - (1 - \epsilon)^k$.

Assuming the uniform probability distributions on $\mathcal{X}^{\leq n}$, for $n \in \mathbb{N}$, and taking the limit $n \to \infty$, there are straightforward notions of **expected (average)** running time and of the **success probability** of a Monte-Carlo or Las-Vegas machine, given as **expectation values**; the success probability is bounded below by the given error bound.

b) The complexity class BPP of languages being decidable in polynomial time with **bounded probability error** is the set of languages possessing a non-deterministic Turing machine $\mathcal{T}$, called a **two-sided Monte-Carlo machine**, running in polynomial time, halting for all $w \in \mathcal{X}^*$, and having an error bound $\frac{1}{2} < \epsilon < 1$ such that $\mathcal{T}$ accepts $w \in \mathcal{L}$ in at least a fraction of ϵ of the branches, and $\mathcal{T}$ rejects $w \notin \mathcal{L}$ in at least a fraction of ϵ of the branches. Hence $\mathsf{BPP} = \mathsf{coBPP}$ and $\mathsf{RP} \cup \mathsf{coRP} \subseteq \mathsf{BPP}$, and it is conjectured that $\mathsf{BPP} \not\subseteq \mathsf{NP}$ holds.

We may fix an error bound $0 < \epsilon_0 < 1$ a priorly, by running $\mathcal{T}$ repeatedly, k times say, and accepting an input if and only if it is accepted by a strict majority of the runs: For $w \in \mathcal{L}$, the set of possible results $\{1, 0\}$ is considered as a probability space, with elementary probabilities ϵ and $1 - \epsilon$, respectively. Running $\mathcal{T}$ repeatedly on $w \in \mathcal{L}$ yields independent choices $x_1, \ldots, x_k \in \{1, 0\}$. Letting $x := \sum_{i=1}^k x_i$, rejection is equivalent to $x \leq \frac{k}{2}$. There are 2^{k-1} sequences $[x_1, \ldots, x_k]$ such that $x \leq \frac{k}{2}$, any of them occurring with probability $\leq \epsilon^{\frac{k}{2}}(1 - \epsilon)^{\frac{k}{2}}$. Hence we get $\mu(x \leq \frac{k}{2}) \leq 2^{k-1}\epsilon^{\frac{k}{2}}(1 - \epsilon)^{\frac{k}{2}} = \frac{1}{2}(1 - (2\epsilon - 1)^2)^{\frac{k}{2}}$, and since

$\frac{1}{2} < \epsilon < 1$ it suffices to choose k large enough such that the latter expression is $\leq \epsilon_0$.

(14.8) Function problems. a) Let $\mathcal{X}$ be an alphabet and let $\mathcal{R} \subseteq \mathcal{X}^* \times \mathcal{X}^*$ be a relation. The **function problem** associated with $\mathcal{R}$ is for given $w \in \mathcal{X}^*$ to find a **solution** $v \in \mathcal{X}^*$ such that $[w, v] \in \mathcal{R}$, if such a v exists at all, otherwise to report failure. A Turing machine $\mathcal{T}$ **solves** the function problem $\mathcal{R}$, if $\mathcal{T}$ halts for all $w \in \mathcal{X}^*$, and outputs a solution if any exists at all, and fails otherwise. There are straightforward notions of running time and complexity classes.

There are straightforward generalisations to non-deterministic Turing machines, in particular there are **Monte-Carlo** and **Las-Vegas machines** for function problems, where the former yield results which might be incorrect with a given error probability, while the latter always yield correct results or with a given error probability report failure.

b) The function problems associated with $\mathcal{L} \in \mathsf{NP}$ are the function problems associated with the polynomial certificates $\mathcal{R}$ for $\mathcal{L}$. Let FNP be the complexity class of function problems associated with languages in NP. In particular, function problems in FNP are solvable by non-deterministic Turing machines running in polynomial time. Let $\mathsf{FP} \subseteq \mathsf{FNP}$ be the complexity class of function problems being solvable by Turing machines running in polynomial time; it is conjectured that $\mathsf{FP} \neq \mathsf{FNP}$ holds.

c) A language $\mathcal{L} \subseteq \mathcal{X}^*$ **polynomial time reduces** to a language $\mathcal{L}' \subseteq \mathcal{X}^*$, if there is a function problem in FP, associated with a relation $\mathcal{R} \subseteq \mathcal{X}^* \times \mathcal{X}^*$, such that for all $w \in \mathcal{X}^*$ there is $v \in \mathcal{X}^*$ such that $[w, v] \in \mathcal{R}$, i. e. failure does not occur, and for all $[w, v] \in \mathcal{R}$ we have $w \in \mathcal{L}$ if and only if $v \in \mathcal{L}'$. Languages $\mathcal{L}$ and $\mathcal{L}'$ are called **polynomial time equivalent**, if $\mathcal{L}$ polynomial time reduces to $\mathcal{L}'$ and vice versa. A Turing machine deciding $\mathcal{L}'$ is called an **oracle** for $\mathcal{L}$. Given a complexity class C of languages, $\mathcal{L}' \in \mathsf{C}$ is called **C-complete** if each $\mathcal{L} \in \mathsf{C}$ polynomial time reduces to $\mathcal{L}'$.

There are straightforward notions of polynomial time reduction, polynomial time equivalence, and oracles for function problems. Given a complexity class C of languages, a function problem is called **C-hard** if each language $\mathcal{L} \in \mathsf{C}$ polynomial time reduces to that function problem.

15 Integer arithmetic

(15.1) Landau symbols. Let $\{n \in \mathbb{N}; n \geq N\} \subseteq D_N \subseteq \mathbb{N}_0$, where $N \in \mathbb{N}_0$. For an **eventually positive function** $f \colon D_N \to \mathbb{R}$, i. e. we have $f(n) > 0$ for all $n \geq N$, let the **Landau symbols** $O(f)$ and $o(f)$ be the sets of all eventually positive functions $g \colon D_N \to \mathbb{R}$ such that the sequence $[\frac{g(n)}{f(n)} \in \mathbb{R}_{>0}; n \geq N]$ is bounded, and such that $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$, respectively; hence $o(f) \subseteq O(f)$. Eventually positive functions $f, g \colon D_N \to \mathbb{R}$ are called **asymptotically equivalent**, $f \sim g$, if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$; in this case $f \in O(g)$ and $g \in O(f)$. We use a similar notation for functions in several variables, and for functions defined on right unbounded subsets of $\mathbb{R}$.

E. g. we have Stirling's formula $\lim_{n\to\infty} \frac{n!\cdot e^n}{n^n\cdot\sqrt{2\pi n}} = 1$, see [6, Formula 96.2], and thus $n! \sim (\frac{n}{e})^n \cdot \sqrt{2\pi n}$, hence $\ln(n!) \sim n(\ln(n)-1) + \frac{1}{2}\cdot\ln(n) + \ln(\sqrt{2\pi})$ and thus $\ln(n!) \sim n\ln(n)$. Letting $\pi(n) := |\{p \in \mathbb{N}; p \leq n, p \text{ prime}\}| \in \mathbb{N}_0$ for $n \in \mathbb{N}$, by the Prime Number Theorem, see [5, Ch.22], we have $\pi(n) \sim \frac{n}{\ln(n)}$.

(15.2) Bit lengths and bit operations. The number of digits to the base $1 \neq z \in \mathbb{N}$ necessary to represent $n = \sum_{i=0}^{b} n_i z^i \in \mathbb{N}$, where $n_i \in \mathbb{Z}_z$, is given as the **bit length** $b_z(n) := 1 + b = 1 + \lfloor \log_z(n) \rfloor = 1 + \lfloor \frac{\ln(n)}{\ln(z)} \rfloor$. For $n \in \mathbb{Z}$ we need an additional sign, hence for the input length of $0 \neq n \in \mathbb{Z}$ we have $1 + b_z(|n|) \in O(\ln(n))$.

The computational complexity of integer arithmetic is counted in **bit operations**, i. e. and, or, exclusive or, not and shift, on bit strings, hence with respect to the base $z = 2$. Generalised bit operations are **Byte operations**, **word operations** and **long word operations**, with respect to the bases $z = 2^8$, $z = 2^{32}$ and $z = 2^{64}$, respectively. The time needed for these operations indeed is polynomial in the input length $1 + b_z(|n|)$.

We treat bit operations as oracles. An algorithm using integer arithmetic, whose input up to sign is $n \in \mathbb{N}$, is called an **$L_{\alpha,c}$-time algorithm**, where $0 \leq \alpha \leq 1$ and $c > 0$, if it needs $L_{\alpha,c} := O(e^{c(\ln(n))^\alpha(\ln(\ln(n)))^{1-\alpha}})$ bit operations. For $\alpha = 0$ we have $L_{\alpha,c} = O(\ln(n)^c)$, thus a polynomial time algorithm. For $\alpha = 1$ we have $L_{\alpha,c} = O(e^{c\ln(n)}) = O(n^c)$, thus an exponential time algorithm. For $0 < \alpha < 1$ we have $cx^\alpha \ln(x)^{1-\alpha} \in o(x)$, thus an **subexponential time** algorithm, i. e. it runs in time $O(e^{h(\ln(n))})$ for some eventually positive function $h(x) \in o(x)$.

(15.3) Ring operations. a) Addition. Let $n \geq m \in \mathbb{N}$ and $b := b_z(n)$, for some $1 \neq z \in \mathbb{N}$. Hence we have $n = \sum_{i=0}^{b-1} n_i z^i$, where $n_i \in \mathbb{Z}_z$, and we may assume $m = \sum_{j=0}^{b-1} m_j z^j$, where $m_j \in \mathbb{Z}_z$, by letting $m_j := 0$ for $j \in \{b_z(m), \ldots, b-1\}$.

```
$\delta := 0 \in \mathbb{Z}_z$
for $k \in [0, \ldots, b-1]$ do
    $s_k := n_k + m_k + \delta \in \mathbb{Z}_{2z}$
    if $s_k \geq z$ then
        $s_k := s_k - z \in \mathbb{Z}_z$
        $\delta := 1 \in \mathbb{Z}_z$
    else $\delta := 0 \in \mathbb{Z}_z$
$s_b := \delta \in \mathbb{Z}_z$
return $[s_0, \ldots, s_b]$
```

Hence we have $n + m = \sum_{k=0}^{b} s_k z^k$. For each k this needs a fixed number of bit operations, and hence needs $O(b_z(n)) = O(\ln(n))$ bit operations; subtraction also needs $O(b_z(n))$ bit operations.

b) Multiplication. Let $n,m \in \mathbb{N}$ and $b_n := b_z(n)$ and $b_m := b_z(m)$, hence $nm = \sum_{i=0}^{b_n-1}\sum_{j=0}^{b_m-1} n_i m_j z^{i+j} = \sum_{k=0}^{b_n+b_m-1}(\sum_{l=\max\{0,k-b_m+1\}}^{\min\{b_n-1,k\}} n_l m_{k-l})z^k$.

```
for $k \in [0,\ldots,b_n+b_m-1]$ do $s_k := 0 \in \mathbb{Z}_z$
for $i \in [0,\ldots,b_n-1]$ do
    $\delta := 0 \in \mathbb{Z}_z$
    for $j \in [0,\ldots,b_m-1]$ do
        $s := s_{i+j} + n_i m_j + \delta \in \mathbb{N}_0$      # $s = (s \bmod z) + \lfloor \frac{s}{z} \rfloor \cdot z$
        $s_{i+j} := (s \bmod z) \in \mathbb{Z}_z$
        $\delta := \lfloor \frac{s}{z} \rfloor \in \mathbb{N}_0$
    $s_{i+b_m} := \delta \in \mathbb{Z}_z$
return $[s_0,\ldots,s_{b_n+b_m-1}]$
```

Hence we have $nm = \sum_{k=0}^{b_n+b_m-1} s_k z^k$. For each i and j this needs a fixed number of bit operations, thus needs $O(b_z(n)b_z(m)) = O(\ln(n)\ln(m))$ bit operations.

(15.4) Quotient and remainder. Let $m \geq n \in \mathbb{N}$, hence there are unique $q,r \in \mathbb{N}_0$ such that $r < n$ and $m = qn + r$.

Let $b' := b_z(m)$ and $b'' := b_z(n)$, for some $1 \neq z \in \mathbb{N}$. Replacing $[m,n]$ by a suitable multiple $[km,kn]$, for some $1 \leq k < z$, we may assume that $n_{b''-1} \geq \lfloor \frac{z}{2} \rfloor$. After replacing n by nz^l for some $l \in \mathbb{N}_0$, i. e. after a suitable shift, we may assume that we have $b_z(n) = b$ and $b_z(m) \in \{b, b+1\}$, where $b \in \{b', b'+1\}$. To compute q, we let $q' := \min\{\lfloor \frac{m_b z + m_{b-1}}{n_{b-1}} \rfloor, z-1\}$. Then we have $q' - 2 \leq q \leq q'$:

We have $n_{b-1}q' \geq m_b z + m_{b-1} - (n_{b-1}-1)$. Hence $m - q'n \leq m - q'n_{b-1}z^{b-1} \leq m - (m_b z + m_{b-1})z^{b-1} + (n_{b-1}-1)z^{b-1} = (n_{b-1}-1)z^{b-1} + \sum_{j=0}^{b-2} m_j z^j < n_{b-1}z^{b-1} \leq n$. From $q \leq z-1$ we conclude $q \leq q'$. We have $q' \leq \frac{m}{n_{b-1}z^{b-1}} < \frac{m}{n-z^{b-1}}$ and $q = \lfloor \frac{m}{n} \rfloor > \frac{m}{n} - 1$. Assume that $3 \leq q' - q < \frac{m}{n-z^{b-1}} - (\frac{m}{n} - 1) = \frac{m \cdot z^{b-1}}{n(n-z^{b-1})} + 1$, then we have $\frac{m}{n} > 2(n_{b-1}-1)$, and hence $z - 4 \geq q' - 3 \geq q = \lfloor \frac{m}{n} \rfloor \geq 2(n_{b-1}-1) \geq z - 3$, a contradiction. Thus $q' - 2 \leq q$. ♯

Computing $[km,kn]$ needs $O(b')$ bit operations, the shifts need $O(b''(b'-b''))$ bit operations, to compute the quotient q at most 3 trials are necessary, since $b_z(q') = 1$ the trial multiplication to compute $q'n$ needs $O(b) = O(b')$ bit operations, and the addition $r := m - qn$ needs $O(b')$ bit operations. This amounts to $O(\max\{b', b''(b'-b'')\})$ bit operations, where $b''(b'-b'') \geq b'$ whenever $b' > b''$, hence since $m \geq n$ this needs $O(\ln(m)\ln(n)) \subseteq O(\ln(m)^2)$ bit operations.

(15.5) Modular exponentiation. Let $e,n \in \mathbb{N}$ and $m \in \mathbb{Z}_n$. We compute $m^e \in \mathbb{Z}/n\mathbb{Z}$ as follows:

```
$r := 1 \in \mathbb{Z}_n$
while $e > 0$ do
    if $(e \bmod 2) = 1$ then $r := (rm \bmod n) \in \mathbb{Z}_n$
    $e := \lfloor \frac{e}{2} \rfloor \in \mathbb{N}_0$
```

```
        $m := (m^2 \bmod n) \in \mathbb{Z}_n$
    return $r$
```

Using the binary representation of $e \in \mathbb{N}$ shows that $r \in \mathbb{Z}_n$ fulfils $r = m^e \in \mathbb{Z}/n\mathbb{Z}$. Since $b_2(e) \in O(\ln(e))$, and multiplication and computing remainders need $O(\ln(n)^2)$ bit operations, we need $O(\ln(e)\ln(n)^2)$ bit operations; conventional exponentiation needs $O(e\ln(n)^2)$ bit operations.

(15.6) Extended Euclidean algorithm. Let $m, n \in \mathbb{N}$. We compute the greatest common divisor $\gcd(m,n) \in \mathbb{N}$ as follows:

```
$r_0 := m \in \mathbb{N}$; $s_0 := 1 \in \mathbb{Z}$; $t_0 := 0 \in \mathbb{Z}$
$r_1 := n \in \mathbb{N}$; $s_1 := 0 \in \mathbb{Z}$; $t_1 := 1 \in \mathbb{Z}$
$i := 1 \in \mathbb{N}$
while $r_i \neq 0$ do
    $r_{i+1} := (r_{i-1} \bmod r_i) \in \mathbb{Z}_{r_i}$
    $q_i := \lfloor \frac{r_{i-1}}{r_i} \rfloor \in \mathbb{N}_0$      # quotient and remainder
    $s_{i+1} := s_{i-1} - q_i s_i \in \mathbb{Z}$
    $t_{i+1} := t_{i-1} - q_i t_i \in \mathbb{Z}$
    $i := i + 1 \in \mathbb{N}$
return $[r_{i-1}, s_{i-1}, t_{i-1}]$
```

We have $r_0 = s_0 m + t_0 n$ and $r_1 = s_1 m + t_1 n$, and by induction on $i \geq 1$ we have $r_{i+1} = r_{i-1} - q_i r_i = (s_{i-1}m + t_{i-1}n) - q_i \cdot (s_i m + t_i n) = s_{i+1}m + t_{i+1}n$. Since $r_{i+1} < r_i$ for $i \geq 1$, the algorithm terminates, after step $i := l+1$ say, returning $[d, s, t] := [r_l, s_l, t_l]$. Thus we have $d = sm + tn$, hence $\gcd(m,n) \mid d$. Conversely, since $r_{l+1} = 0$, for $i \in \{l, l-1, \ldots, 1\}$ we by induction have $r_l \mid r_{i+1}, r_i$ and thus $r_l \mid q_i r_i + r_{i+1} = r_{i-1}$, hence in particular $r_l \mid r_0, r_1$, thus $r_l \mid \gcd(m,n)$. Thus we have $0 < d = \gcd(m,n) = sm + tn$ with Bézout coefficients s and t; if the latter are not needed, the computation of the s_i and t_i can be left out.

Let $1 \neq z \in \mathbb{N}$. For $i \in \{1, \ldots, l\}$ we need $O(b_z(r_i) b_z(q_i))$ bit operations to compute $[q_i, r_i]$. Since $b_z(q_i) = 1 + \lfloor \log_z(q_i) \rfloor$, we have $O(\sum_{i=1}^l b_z(q_i)) = O(b_z(\prod_{i=1}^l q_i)) \subseteq O(b_z(r_0))$. Hence computing the quotients and remainders needs $O(\sum_{i=1}^l b_z(r_i) b_z(q_i)) \subseteq O(b_z(r_1) \cdot \sum_{i=1}^l b_z(q_i)) \subseteq O(b_z(r_1) b_z(r_0))$ bit operations. To compute the linear combination needs $O(\sum_{i=1}^l b_z(q_i) b_z(s_i))$ bit operations, where in turn $b_z(s_i) \in O(b_z(s_{i-1}) + b_z(q_{i-1}))$, hence we have $b_z(s_i) \in O(\sum_{j=1}^{i-1} b_z(q_j))$, yielding $O(\sum_{i=1}^l \sum_{j=1}^{i-1} b_z(q_i) b_z(q_j))$ bit operations. As above we from this obtain $O(\sum_{j=1}^{l-1} \sum_{i=j+1}^{l} b_z(q_j) b_z(q_i)) \subseteq O(\sum_{j=1}^{l-1} b_z(q_j) b_z(r_j)) \subseteq O(b_z(r_1) \cdot \sum_{j=1}^{l-1} b_z(q_j)) \subseteq O(b_z(r_1) b_z(r_0))$ bit operations. Thus this needs $O(b_z(r_1) b_z(r_0)) = O(b_z(m) b_z(n))$ bit operations; if $m \geq n$ this hence needs $O(\ln(m)^2)$ bit operations.

16 Primality testing

(16.1) Fermat-Lucas test [1876]. Let $1 \neq n \in \mathbb{N}$. Then n is a prime if and only if $(\mathbb{Z}/n\mathbb{Z})^*$ is cyclic of order $n-1$: If n is a prime then $\mathbb{Z}/n\mathbb{Z}$ is a field and hence $(\mathbb{Z}/n\mathbb{Z})^* \cong C_{n-1}$; conversely, if $\varphi(n) = |(\mathbb{Z}/n\mathbb{Z})^*| = n-1$ then n is a prime. To verify this condition we specify a primitive root $x \in (\mathbb{Z}/n\mathbb{Z})^*$: Let $n-1 = \prod_{i=1}^{r} p_i^{a_i}$, where the $p_i \in \mathbb{N}$ are pairwise distinct primes and $a_i \in \mathbb{N}$. Then x has order $n-1$ if and only if $x^{n-1} = 1$ and $x^{\frac{n-1}{p_i}} \neq 1$ for $i \in \{1, \ldots, r\}$. If n is a prime, then the tuple $[x; p_1, \ldots, p_r]$ as above is a called a **Lucas primality witness** for n. If n is a prime then for all $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $x^{n-1} = 1$. Hence an element $x \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $x^{n-1} \neq 1$ is called a **Fermat compositeness witness** for n.

E. g. for $n \in \mathbb{N}_0$ let $F_n := 2^{2^n} + 1 \in \mathbb{N}$ be the n-th **Fermat number**, where $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65537$ are primes, with Lucas witnesses $[2;2]$, $[3;2]$, $[3;2]$, $[3;2]$, $[3;2]$, respectively. It was conjectured [Fermat, 1640] that F_n always is a prime, but $3^{F_5-1} = 3^{2^{32}} \equiv 3\,029\,026\,160 \not\equiv 1 \pmod{F_5}$, hence $F_5 = 4\,294\,967\,297$ is composite. All F_n for $n \in \{5, \ldots, 30\}$ are known to be composite; and Pepin's test [1877], proved using the quadratic reciprocity law for Jacobi symbols, says that for $n \geq 1$ the element 3 always is a Lucas primality or Fermat compositeness witness, see [7, Exc.18.25].

If n is composite and $x^{n-1} = 1$ for $1 \neq x \in (\mathbb{Z}/n\mathbb{Z})^*$, then n is called a **Fermat pseudoprime** with respect to the **base** x, the latter is called a **Fermat liar** for n. Then $U_n := \{x \in (\mathbb{Z}/n\mathbb{Z})^*; x^{n-1} = 1\} \leq (\mathbb{Z}/n\mathbb{Z})^*$ is a subgroup. Hence if $U_n \neq (\mathbb{Z}/n\mathbb{Z})^*$ then we have $|U_n| \leq \frac{|(\mathbb{Z}/n\mathbb{Z})^*|}{2}$, implying that in this case the fraction of compositeness witnesses is at least $\frac{1}{2}$.

If n is a Fermat pseudoprime with respect to all bases $1 \neq x \in (\mathbb{Z}/n\mathbb{Z})^*$, i. e. we have $U_n = (\mathbb{Z}/n\mathbb{Z})^*$, then n is called a **Carmichael number** [Korselt, 1899; Carmichael, 1910]. We have $n^{\frac{2}{7}} \leq |\{k \in \{1, \ldots, n\}; k \text{ Carmichael number}\}| \leq n^{1-(1+o(1)) \cdot \frac{\ln\ln\ln(n)}{\ln\ln(n)}}$ for $n >> 0$ [Alford-Granville-Pomerance, 1992; Pomerance-Selfridge-Wagstaff, 1980], hence there are infinitely many Carmichael numbers.

(16.2) Proposition. Let $p \in \mathbb{N}$ be an odd prime, and let $a \in \mathbb{N}$. Then we have $(\mathbb{Z}/p^a\mathbb{Z})^* \cong C_{p^{a-1}(p-1)}$.

Proof. Since $(\mathbb{Z}/p\mathbb{Z})^* \cong C_{p-1}$ there is $x \in (\mathbb{Z}/p^a\mathbb{Z})^*$ such that $\overline{x} \in (\mathbb{Z}/p\mathbb{Z})^*$ has order $p-1$, hence x has order divisible by $p-1$, and thus we may assume that x has order $p-1$. Assume there is $y \in (\mathbb{Z}/p^a\mathbb{Z})^*$ having order p^{a-1}, then $xy \in (\mathbb{Z}/p^a\mathbb{Z})^*$ has order $p^{a-1}(p-1)$. Since $|(\mathbb{Z}/p^a\mathbb{Z})^*| = \varphi(p^a) = p^{a-1}(p-1)$ this implies that $(\mathbb{Z}/p^a\mathbb{Z})^*$ is cyclic. By induction on $a \in \mathbb{N}$ we show that $(1+p)^{p^{a-1}} = 1 + k_a p^a$ where $p \not| \, k_a \in \mathbb{N}$, this implies that $y := 1 + p \in \mathbb{Z}$ has order p^{a-1} in $(\mathbb{Z}/p^a\mathbb{Z})^*$:

The case $a = 1$ is fulfilled with $k_a = 1$. For $a \geq 2$ we have $(1+p)^{p^{a-1}} = (1 + k_{a-1}p^{a-1})^p = 1 + k_{a-1}p^a + \frac{p(p-1)}{2} k_{a-1}^2 p^{2(a-1)} + \sum_{i=3}^{p} \binom{p}{i} k_{a-1}^i p^{i(a-1)}$. We have $2(a-1)+1 \geq a+1$ if and only if $a \geq 2$, and $i(a-1) \geq a+1$ if and only if $a \geq \frac{i+1}{i-1} = 1 + \frac{2}{i-1}$, which for $a \geq 2$ and $i \geq 3$ is fulfilled. Thus we have $(1+p)^{p^{a-1}} = 1 + k_a p^a$ where $k_a \equiv k_{a-1} \pmod{p}$. ♯

(16.3) Proposition. Let $p \in \mathbb{N}$ be an odd prime, and let $p-1 = 2^l m$, where $l \in \mathbb{N}$ and m is odd. Then for all $x \in (\mathbb{Z}/p\mathbb{Z})^*$ we have either $x^m = 1$ or there is a unique $k \in \{0, \ldots, l-1\}$ such that $x^{2^k m} = -1$.

Proof. The conditions are mutually exclusive. Let $x \in (\mathbb{Z}/p\mathbb{Z})^*$ such that $x^m \neq 1$, and let $\mathcal{J} := \{j \in \{0, \ldots, l-1\}; x^{2^j m} \neq 1\}$. We have $0 \in \mathcal{J} \neq \emptyset$, hence let $k := \max \mathcal{J}$ and $y := x^{2^k m} \in (\mathbb{Z}/p\mathbb{Z})^*$. Using $x^{p-1} = x^{2^l m} = 1$, this implies $y^2 = 1$, and thus y is a root of $X^2 - 1 \in \mathbb{Z}/p\mathbb{Z}[X]$. Since $\mathbb{Z}/p\mathbb{Z}$ is a field and $y \neq 1$ we conclude $y = -1$. ♯

(16.4) Theorem: [Miller, 1976; Rabin, 1980]. Let $9 \neq n \in \mathbb{N}$ be odd and composite, and let $n-1 = 2^l m$, where $l \in \mathbb{N}$ and m is odd. Then the fraction of elements $x \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $x^m = 1$ or $x^{2^k m} = -1$ for some $k \in \{0, \ldots, l-1\}$ is at most $\frac{1}{4}$.

Proof. Let $B_n^* := \{x \in (\mathbb{Z}/n\mathbb{Z})^*; x^m = 1\}$, let $B_{n,k} := \{x \in (\mathbb{Z}/n\mathbb{Z})^*; x^{2^k m} = -1\}$ for $k \in \{0, \ldots, l-1\}$, and let $B_n := B_n^* \,\dot\cup\, \coprod_{k=0}^{l-1} B_{n,k} \subseteq U_n := \{x \in (\mathbb{Z}/n\mathbb{Z})^*; x^{n-1} = 1\}$. We have to show that $|B_n| \leq \frac{1}{4}\varphi(n)$ holds:

Let $\varphi(p_i^{a_i}) = p_i^{a_i-1}(p_i - 1) = 2^{l_i} m_i$, where $l_i, m_i \in \mathbb{N}$ and m_i is odd; we may assume that $l_1 \leq l_2 \leq \cdots \leq l_r$. Hence we have $\varphi(n) = \prod_{i=1}^r \varphi(p_i^{a_i}) = \prod_{i=1}^r 2^{l_i} m_i$ and $\gcd(n-1, \varphi(p_i^{a_i})) = 2^{l'_i} m'_i$, where $l'_i = \min\{l, l_i\}$ and $m'_i = \gcd(m, m_i)$.

We have $B_n^* \cong \prod_{i=1}^r \{x \in (\mathbb{Z}/p_i^{a_i}\mathbb{Z})^*; x^m = 1\}$ as groups. Since $(\mathbb{Z}/p_i^{a_i}\mathbb{Z})^* \cong C_{\varphi(p^{a_i})}$ we conclude $|B_n^*| = \prod_{i=1}^r \gcd(m, \varphi(p_i^{a_i})) = \prod_{i=1}^r m'_i$. We have $B_{n,k} \cong \prod_{i=1}^r \{x \in (\mathbb{Z}/p_i^{a_i}\mathbb{Z})^*; x^{2^k m} = -1\}$ as sets, thus for $k \geq l_1$ we have $B_{n,k} = \emptyset$. For $k < l_1$ we have $|\{x \in (\mathbb{Z}/p_i^{a_i}\mathbb{Z})^*; x^{2^k m} = -1\}| = |\{x \in (\mathbb{Z}/p_i^{a_i}\mathbb{Z})^*; x^{2^{k+1} m} = 1\}| - |\{x \in (\mathbb{Z}/p_i^{a_i}\mathbb{Z})^*; x^{2^k m} = 1\}| = \gcd(2^{k+1} m, \varphi(p_i^{a_i})) - \gcd(2^k m, \varphi(p_i^{a_i})) = (2^{k+1} - 2^k)\gcd(m, m_i) = 2^k m'_i$, implying $|B_{n,k}| = 2^{kr} \prod_{i=1}^r m'_i$. Hence $|B_n| = \prod_{i=1}^r m'_i + \sum_{k=0}^{l_1-1} (2^{kr} \prod_{i=1}^r m'_i) = (1 + \frac{2^{rl_1}-1}{2^r-1}) \cdot \prod_{i=1}^r m'_i$, thus $|B_n| = \alpha\beta\varphi(n)$, where $\alpha := \frac{2^{rl_1}+2^r-2}{(2^r-1) \cdot 2^{\sum_{i=1}^r l_i}}$ and $\beta := \prod_{i=1}^r \frac{m'_i}{m_i} \leq 1$.

Since $\alpha \leq \frac{1}{2^r-1} \cdot (1 + \frac{2^r-2}{2^{rl_1}}) \leq \frac{1}{2^r-1} \cdot (1 + \frac{2^r-2}{2^r}) = \frac{2(2^r-1)}{2^r(2^r-1)} = \frac{1}{2^{r-1}}$ we are done for $r \geq 3$. If $r = 2$ and $[m_1, m_2] \neq [m'_1, m'_2]$, then we have $\alpha \leq \frac{1}{2}$ and $\beta \leq \frac{1}{2}$ and we are done as well. Hence let $r = 2$ and $m_i = m'_i$. Then we have $p_i^{a_i-1} \mid m_i = m'_i \mid m \mid n-1$, and from $p_i \mid n$ we conclude $a_i = 1$, and thus $n = p_1 p_2$. Since $a_i = 1$ we have $p_i - 1 = 2^{l_i} m_i$, and hence $0 \equiv n-1 \equiv p_1 p_2 - 1 \equiv p_j - 1 \equiv 2^{l_j} m_j \pmod{m_i}$ for $j \neq i$. This implies $m_i \mid m_j$, and hence we have $m_1 = m_2$ and $l_1 < l_2$. Thus we obtain $\alpha \leq \frac{1}{2^2-1} \cdot \frac{2^{2l_1}+2^2-2}{2^{l_1+l_2}} \leq \frac{1}{3}(\frac{2^{2l_1}}{2^{2l_1+1}} + \frac{2}{2^3}) = \frac{1}{3}(\frac{1}{2} + \frac{1}{4}) = \frac{1}{4}$.

Table 9: Strong pseudoprimes

t	$p_1, \ldots, p_t$	n	factorization	$\sim$
1	2	2047	$23{\cdot}89$	$2{\cdot}10^3$
2	$2, 3$	1373653	$829{\cdot}1657$	$1{\cdot}10^6$
3	$2, 3, 5$	25326001	$2251{\cdot}11251$	$3{\cdot}10^7$
4	$2, 3, 5, 7$	3215031751	$151{\cdot}751{\cdot}28351$	$3{\cdot}10^9$
5	$2, 3, 5, 7, 11$	2152302898747	$6763{\cdot}10627{\cdot}29947$	$2{\cdot}10^{12}$
6	$2, 3, 5, 7, 11, 13$	3474749660383	$1303{\cdot}16927{\cdot}157543$	$3{\cdot}10^{12}$
7	$2, 3, 5, 7, 11, 13, 17$	341550071728321	$10670053{\cdot}32010157$	$3{\cdot}10^{15}$
8	$2, 3, 5, 7, 11, 13, 17, 19$	341550071728321	$10670053{\cdot}32010157$	$3{\cdot}10^{15}$

Finally let $r = 1$ and $a_1 > 1$. Since $(\mathbb{Z}/p_1^{a_1}\mathbb{Z})^*$ is cyclic, we have $|B_n| \leq |U_n| = \gcd(\varphi(n), n-1) = \gcd(p_1^{a_1-1}(p_1-1), p_1^{a_1}-1) = p_1 - 1$, and thus we obtain $\frac{|B_n|}{\varphi(n)} \leq \frac{p_1-1}{p_1^{a_1-1}(p_1-1)} = \frac{1}{p_1^{a_1-1}} \leq \frac{1}{4}$. $\sharp$

(16.5) Miller-Rabin test. Let $1 \neq n \in \mathbb{N}$ be odd. An element $x \in (\mathbb{Z}/n\mathbb{Z})^* \setminus B_n$ is called a **strong compositeness witness** for n, hence for composite $n \neq 9$ the fraction of compositeness witnesses amongst all elements of $(\mathbb{Z}/n\mathbb{Z})^*$ is at least $\frac{3}{4}$; for $n = 9$ we have $B_9 = \{\pm 1\}$, while $\varphi(9) = 6$ and hence $(\mathbb{Z}/9\mathbb{Z})^* \cong C_6$. Since $l \in O(\ln(n))$ and exponentiation needs $O(\ln(n)^3)$ bit operations, a strong compositeness test needs $O(\ln(n)^4)$ bit operations. This yields a polynomial time Monte-Carlo algorithm to prove compositeness, which actually is the workhorse of modern primality testing.

If n is composite and $x \in B_n$, then n is called a **strong pseudoprime** with respect to the **base** x, the latter is called a **strong liar** for n; in this we have $x^{n-1} = x^{2^l m} = 1$, hence x also is a Fermat liar. Although there are composite $n \neq 9$ which are strong pseudoprimes with respect to a fraction of $\frac{1}{4}$ of the bases, for most n this fraction is much smaller. Still, it is possible to construct strong pseudoprimes for any given finite set of bases [Arnault, 1995].

We have $B_n^* \leq (\mathbb{Z}/n\mathbb{Z})^*$, and for $k, k' \in \{0, \ldots, l-1\}$ such that $k > k'$ we have $B_n^* B_{n,k} = B_{n,k}$ and $B_{n,k}B_{n,k'} \subseteq B_{n,k}$, as well as $B_{n,0}B_{n,0} \subseteq B_n^*$, but in general $B_{n,k}B_{n,k} \not\subseteq B_{n,k-1}$. Thus if n is a strong pseudoprime with respect to the bases $x, y \in (\mathbb{Z}/n\mathbb{Z})^*$, then n is not necessarily a strong pseudoprime with respect to the base xy, but still this seems to be likely. Hence we are tempted to only consider bases $x = \overline{z} \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $z \in \mathbb{N}$ can be chosen to be prime: Let $[p_1, \ldots, p_t] \subseteq \mathbb{N}$ be the sequence of the first $t \in \mathbb{N}$ primes. Then the smallest composite odd $n \in \mathbb{N}$ which is a strong pseudoprime with respect to the bases $\{p_1, \ldots, p_t\}$ for $t \in \{1, \ldots, 8\}$ are given in Table 9.

(16.6) Other primality tests and complexity. Let PRIMES be the following decision problem: Given $1 \neq n \in \mathbb{N}$, is n prime? Hence the complementary decision problem is $\mathsf{COMPOSITES}$: Given $1 \neq n \in \mathbb{N}$, is n composite? The classical result (16.7) shows that PRIMES is in $\mathsf{NP} \cap \mathsf{coNP}$.

The Solovay-Strassen test [1977], based on Euler's criterion for being a square and using Jacobi symbols, also is a polynomial time Monte-Carlo algorithm to prove compositeness, the associated liars being called **Euler liars**. But since it is more expensive and at the same time has more liars, it is superseded by the Rabin-Miller test, see [10, Ch.4.2]. Anyway, this shows that PRIMES is in coRP.

Adleman-Huang [1992] have given a polynomial time Monte-Carlo algorithm to prove primality; this algorithm uses hyperelliptic curves of genus 2 and is impractical, but it shows that PRIMES is in RP and thus in ZPP. For practical purposes the Elliptic Curve Primality Proving (ECPP) algorithm [Atkin-Morain, 1990] is used, which is based on the impractical Goldwasser-Kilian test [1986]. The ECPP algorithm needs expected polynomial time, but in the worst case might be much slower. The largest integers proven to be prime have size $\sim 10^{1000}$; see [10, Ch.4.7] and [7, Ch.18.6] and [3, Ch.9].

The Jacobi sum test [Adleman-Pomerance-Rumely, 1983] to deterministically decide primality of n runs in time $O(\ln(n)^{c \ln\ln\ln(n)})$, which is quite close to polynomial time. Finally, Agrawal-Kayal-Saxena [2002] have given an astonishingly simple polynomial time algorithm to decide primality; this algorithm is as yet impractical, but shows that PRIMES is in P.

(16.7) Theorem: Pratt [1975]. PRIMES is in $\mathsf{NP} \cap \mathsf{coNP}$.

Proof. a) We provide a polynomial certificate for being composite: Let $\mathcal{R} := \{[n, n'] \in \mathbb{N} \times \mathbb{N}; 1 < n' < n, n' \mid n\}$. Hence $1 \neq n \in \mathbb{N}$ is composite if and only if there is $n' \in \mathbb{N}$ such that $[n, n'] \in \mathcal{R}$. If $[n, n'] \in \mathcal{R}$, then for the bit lengths we have $\ln(n') \leq \ln(n)$. For $[n, n'] \in \mathbb{N} \times \mathbb{N}$ we can decide whether $n' \mid n$ using $O(\ln(n') \ln(n))$ bit operations, where the bit length of $[n, n']$ is $\ln(n') + \ln(n)$, hence this decision problem is in P.
b) We provide a polynomial certificate for being a prime: Since $1 \neq n \in \mathbb{N}$ is a prime if and only if there is a Lucas witness for n, we let $\mathcal{R}' := \{[n; x, p_1, \ldots, p_r]; [x, p_1, \ldots, p_r] \text{ Lucas witness for } n\}$. If $[n; x, p_1, \ldots, p_r] \in \mathcal{R}'$, then for the bit lengths we have $\ln(x) \leq \ln(n)$ and $\ln(p_i) \leq \ln(n)$, where $r \in O(\ln(n))$. Hence the bit length of a Lucas witness for n is bounded quadratically in $\ln(n)$. To check the defining conditions of Lucas witnesses, we need $O(\ln(n)^2)$ bit operations to compute $\frac{n-1}{p_i}$, and $O(\ln(n)^3)$ bit operations to compute $x^{\frac{n-1}{p_i}} \in \mathbb{Z}/n\mathbb{Z}$ and $x^{n-1} \in \mathbb{Z}/n\mathbb{Z}$. Since $r \in O(\ln(n))$, this amounts to $O(\ln(n)^4)$ bit operations, hence this decision problem is in P. $\sharp$

17 Factorisation

(17.1) The ρ method [Pollard, 1975]. Let $n \in \mathbb{N}$ be composite, let $x_0 \in \mathbb{Z}/n\mathbb{Z}$, and for a function $f\colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ let $x_i := f(x_{i-1})$ for $i \in \mathbb{N}$. We assume that f is chosen such that the $x_i \in \mathbb{Z}/n\mathbb{Z}$ behave like random choices in $\mathbb{Z}/n\mathbb{Z}$. To minimise the number of operations needed, and since linear functions do not behave randomly, in practice functions $f_c\colon x \mapsto x^2 + c$ are used, where $c \in \mathbb{Z}/n\mathbb{Z}$ such that $c \neq 0, -2$, typically $c = \pm 1$; the function f_0 does not behave randomly, and for $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we get $f_{-2}(x + x^{-1}) = x^2 + x^{-2}$.

Let $1 \neq p \in \mathbb{N}$ be a divisor of n. Hence there are $k \in \mathbb{N}_0$ and $l \in \mathbb{N}$ minimal such that we have a **collision** $x_k = x_{k+l} \in \mathbb{Z}/p\mathbb{Z}$; hence $x_k = x_{k+jl} \in \mathbb{Z}/p\mathbb{Z}$ for all $j \in \mathbb{N}_0$, which is the name-giving property of the method. Thus we have $p \mid \gcd(x_k - x_{k+l}, n)$. Assume that $\frac{n}{p} \neq 1$ and $\gcd(p, \frac{n}{p}) = 1$; this is feasible if and only if n is not a prime power. By the Chinese remainder theorem $x_i \in \mathbb{Z}/p\mathbb{Z}$ and $x_i \in \mathbb{Z}/q\mathbb{Z}$ behave like independent random choices in $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/q\mathbb{Z}$, respectively, hence it is likely that $q \not| \gcd(x_k - x_{k+l}, n)$, implying $1 < \gcd(x_k - x_{k+l}, n) < n$, yielding a proper divisor of n.

The number of steps needed until we arrive at a collision is of size $O(\sqrt{p}) = O(\sqrt[4]{n})$, showing that we have a Las-Vegas algorithm to factor n having exponential running time $O(e^{(\frac{1}{4}+o(1))\ln(n)})$: For $t \in \mathbb{N}_0$ such that $t \leq p$ there are p^{t+1} sequences $[x_0, \ldots, x_t] \in (\mathbb{Z}/p\mathbb{Z})^{t+1}$, where precisely $\prod_{i=0}^{t}(p - i)$ of them pairwise different entries. For the exponential function we for $0 \leq \lambda \leq 1$ have $e^{-\lambda} \geq 1 - \lambda \geq 0$, where $|e^{-\lambda} - (1 - \lambda)| \leq \frac{\lambda^2}{2}$. Hence for the fraction of the sequences with pairwise different entries amongst all sequences we get $\prod_{i=0}^{t}(1 - \frac{i}{p}) \leq \prod_{i=0}^{t} e^{-\frac{i}{p}} = e^{\frac{-t(t+1)}{2p}} \leq e^{\frac{-t^2}{2p}}$. Given $0 < \epsilon < 1$, we have $e^{\frac{-t^2}{2p}} < \epsilon$ if and only if $t > \sqrt{-2p\ln(\epsilon)}$; since for $\epsilon = \frac{1}{2}$ and $p = 365$ this yields $t \geq 23$, this is called the **birthday paradox**.

To detect a collision, we can avoid to store all the values $x_0, x_1, \ldots$ successively computed, without increasing the number of steps needed, by using **Floyd's cycle detection trick**: Let $y_0 := x_0 \in \mathbb{Z}/n\mathbb{Z}$ and $y_i := f(f(y_{i-1})) = x_{2i} \in \mathbb{Z}/n\mathbb{Z}$ for $i \in \mathbb{N}$. Then we have $x_i = y_i \in \mathbb{Z}/p\mathbb{Z}$ if and only if $i \geq k$ and $l \mid 2i - i = i$. The minimal $i \in \mathbb{N}$ fulfilling these conditions is an element of $\{k, \ldots, k + l\}$, hence we need at most $k + l$ steps to arrive at the collision $x_i = y_i \in \mathbb{Z}/p\mathbb{Z}$. This yields the following algorithm to factor n, where $f\colon \mathbb{Z}_n \to \mathbb{Z}_n$:

```
choose $x \in \mathbb{Z}_n$ randomly
$y := x \in \mathbb{Z}_n$
$i := 0 \in \mathbb{N}_0$
while $i < n$ do
    $i := i + 1 \in \mathbb{N}$
    $x := f(x) \in \mathbb{Z}_n$
    $y := f(f(y)) \in \mathbb{Z}_n$
    $g := \gcd(x - y, n) \in \mathbb{Z}_n$
    if $1 < g < n$ then return $g$
```

```
od
return fail
```

(17.2) The $p-1$ method [Pollard, 1974]. Let $n \in \mathbb{N}$ be composite. For a bound $B \in \{2, \ldots, n\}$ we for a number of tries do the following:

```
choose $x \in \mathbb{Z}_n$ randomly
if $x = 0$ then return fail
$g := \gcd(x, n) \in \mathbb{Z}_n$
if $1 < g < n$ then return g
$l := 1 \in \mathbb{N}$
for $k \in \{2, \ldots, B\}$ do
    compute $e \in \mathbb{N}_0$ maximal such that $k^e \leq n$
    $l := \mathrm{lcm}(l, k^e) \in \mathbb{N}$
$y := (x^l \bmod n) \in \mathbb{Z}_n$
$g := \gcd(y - 1, n) \in \mathbb{Z}_n$
if $1 < g < n$ then return g
return fail
```

If there is a prime $p \mid n$ such that $p-1$ is **B-smooth**, i. e. for all primes $q \mid p-1$ we have $q \leq B$, then $p-1 \mid l$. For $x \in \mathbb{Z}_n^*$ we have $x \in (\mathbb{Z}/p\mathbb{Z})^*$, and hence from $x^{p-1} = 1 \in (\mathbb{Z}/p\mathbb{Z})^*$ we get $y = x^l = 1 \in (\mathbb{Z}/p\mathbb{Z})^*$, implying $g > 1$. We have $g < n$ if there is a prime $p \neq q \mid n$ such that $x^l \neq 1 \in (\mathbb{Z}/q\mathbb{Z})^*$, i. e. l is not a multiple of the order of $x \in (\mathbb{Z}/q\mathbb{Z})^*$, which is likely to happen if $q-1$ is not B-smooth.

(17.3) Example. We apply the ρ method and the $p-1$ method to a few Fermat numbers; for the ρ method, which in the considered cases always works, we use $c := 1$ and $x_0 := 1$, the parameters for the $p-1$ method in the successful cases are given below:

For $F_5 := 2^{2^5} + 1 = 4\,294\,967\,297 \sim 4 \cdot 10^9$, using the ρ method, or using $B := 5$ and $x := 3$ in the $p-1$ method, we find the prime divisor $p_3 := 641 \mid F_5$, and $p_7 := \frac{F_5}{p_3} = 6\,700\,417$ [Euler, 1732]. We have $p_3 - 1 = 2^7 \cdot 5$, as well as $p_7 - 1 = 2^7 \cdot 3 \cdot 17449$, and $17449 - 1 = 2^3 \cdot 3 \cdot 727$. The Lucas test yields the primality witnesses 17 for 17449, and 5 for p_7.

For $F_6 := 2^{2^6} + 1 \sim 1.8 \cdot 10^{19}$, using the ρ method, or using $B := 17$ and $x := 3$ in the $p-1$ method, we find $p_6 := 274177 \mid F_6$ and $p_{14} := \frac{F_6}{p_6} = 67\,280\,421\,310\,721$ [Landry, 1880]. We have $p_6 - 1 = 2^8 \cdot 3^2 \cdot 7 \cdot 17$, and the Lucas test yields the primality witness 5 for p_6. We have $p_{14} - 1 = 2^8 \cdot 5 \cdot 47 \cdot 373 \cdot 2\,998\,279$, and $2\,998\,279 - 1 = 2 \cdot 3^2 \cdot 166571$, and $166571 - 1 = 2 \cdot 5 \cdot 16657$, and $16657 - 1 = 2^4 \cdot 3 \cdot 347$, and the Lucas test yields the primality witnesses 5 for 16657, and 2 for 166571, and 3 for $2\,998\,279$, and 3 for p_{14}.

For $F_7 := 2^{2^7} + 1 \sim 3.4 \cdot 10^{38}$ we find for which the $p-1$ method is unsuccessful, by the ρ method we find $p_{17} := 59\,649\,589\,127\,497\,217$ and $p_{22} := \frac{F_7}{p_{17}} = 5\,704\,689\,200\,685\,129\,054\,721$. We have $p_{17} - 1 = 2^9 \cdot 116\,503\,103\,764\,643$, and

$116\,503\,103\,764\,643 - 1 = 2 \cdot 7 \cdot 449 \cdot 18\,533\,742\,247$, and $18\,533\,742\,247 - 1 = 2 \cdot 3^3 \cdot 181 \cdot 1896229$, and the Lucas test yields the primality witnesses 11 for $18\,533\,742\,247$, and 2 for $116\,503\,103\,764\,643$, and 3 for p_{17}. We have $p_{22} - 1 = 2^9 \cdot 3^5 \cdot 5 \cdot 12497 \cdot 733\,803\,839\,347$, and $733\,803\,839\,347 - 1 = 2 \cdot 3 \cdot 2203 \cdot 55\,515\,497$, and $55\,515\,497 - 1 = 2^3 \cdot 6939437$, and the Lucas test yields the primality witnesses 3 for $55\,515\,497$, and 2 for $733\,803\,839\,347$, and 23 for p_{22}.

For $F_8 := 2^{2^8} + 1 \sim 10^{77}$, for which the $p-1$ method is unsuccessful, by the ρ method we find $p_{16} := 1\,238\,926\,361\,552\,897$ and $p_{62} := \frac{F_8}{p_{16}}$. We have $p_{16} - 1 = 2^{11} \cdot 157 \cdot 3\,853\,149\,761$, and $3\,853\,149\,761 - 1 = 2^6 \cdot 5 \cdot 719 \cdot 16747$, and the Lucas test yields the primality witnesses 7 for $3\,853\,149\,761$, and 3 for p_{16}. We have $p_{62} - 1 = 2^{11} \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot n$, where $n \sim 3 \cdot 10^{55}$, and by the ρ method we find $n = 31\,618\,624\,099\,079 \cdot n'$ where $n' \sim 10^{42}$. We have $31\,618\,624\,099\,079 - 1 = 2 \cdot 1789 \cdot 10079 \cdot 876769$, and the Lucas test yields the primality witness 17 for $31\,618\,624\,099\,079$. Again by the ρ method we find successively $n' - 1 = 2^4 \cdot 3 \cdot 8861 \cdot 10\,608\,557 \cdot 25\,353\,082\,741\,699 \cdot 9\,243\,081\,088\,796\,207$, and $25\,353\,082\,741\,699 - 1 = 2 \cdot 3^2 \cdot 16879 \cdot 83\,447\,159$, and $83\,447\,159 - 1 = 2 \cdot 41\,723\,579$, and $41\,723\,579 - 1 = 2 \cdot 13 \cdot 1604753$, and the Lucas test yields the primality witnesses 2 for $41\,723\,579$, and 11 for $83\,447\,159$, and 2 for $25\,353\,082\,741\,699$, as well as $9\,243\,081\,088\,796\,207 - 1 = 2 \cdot 20939 \cdot 220\,714\,482\,277$, and $220\,714\,482\,277 - 1 = 2^2 \cdot 3^2 \cdot 6\,130\,957\,841$, and $6\,130\,957\,841 - 1 = 2^4 \cdot 5 \cdot 7 \cdot 10\,948\,139$, and $10\,948\,139 - 1 = 2 \cdot 23 \cdot 29^2 \cdot 283$, and the Lucas test yields the primality witnesses 2 for $10\,948\,139$, and 3 for $6\,130\,957\,841$, and 5 for $220\,714\,482\,277$, and 5 for $9\,243\,081\,088\,796\,207$, which yields the primality witness 11 for n'.

For $F_9 := 2^{2^9} + 1 \sim 1.3 \cdot 10^{154}$, using the ρ method, or using $B := 37$ and $x := 3$ in the $p-1$ method, we find $p_7 := 2\,424\,833 \mid F_9$. We have $p_7 - 1 = 2^{16} \cdot 37$, and the Lucas test yields the primality witness 3 for p_7. For $n := \frac{F_6}{p_6} \sim 5.5 \cdot 10^{146}$ the Fermat test yields the compositeness witness 3; we have $n = p_{49} \cdot p_{99}$, the prime factors having the indicated number of digits.

The Fermat numbers $F_7, F_8, F_9, F_{10}, F_{11}$ have been completely factored by the continued fraction method [Morrison-Brillhart, 1971], the ρ method [Pollard, 1975], the Number Field Sieve [Lenstra, 1990], the Elliptic Curve Method [Brent, 1995], and the Elliptic Curve Method [Brent, 1988], respectively; we have $F_{10} \sim 10^{309}$ and $F_{11} \sim 10^{617}$. In general, already integers of size $\sim 10^{200}$ pose severe problems to factorisation methods.

(17.4) Index calculus methods. Let $n \in \mathbb{N}$ be odd and composite. If $x, y \in \mathbb{Z}_n^*$ such that $x \neq \pm y \in (\mathbb{Z}/n\mathbb{Z})^*$ and $x^2 = y^2 \in (\mathbb{Z}/n\mathbb{Z})^*$, then we have $(x+y)(x-y) = x^2 - y^2 = kn$ for some $k \in \mathbb{Z}$, and thus $1 < \gcd(x+y, n) < n$ and $1 < \gcd(x-y, n) < n$, yielding a factorisation of n. Hence by the **Fermat-Legendre method** simply choose $k \in \mathbb{N}$ small, and for increasing $x > \lfloor \sqrt{kn} \rfloor$ check whether $x^2 - kn \in \mathbb{N}$ is a square.

If n has $r \in \mathbb{N}$ distinct prime divisors, then by the Chinese remainder theorem and the cyclicity of $(\mathbb{Z}/p^a\mathbb{Z})^*$, where $p \in \mathbb{N}$ is an odd prime and $a \in \mathbb{N}$, we conclude that any square in $(\mathbb{Z}/n\mathbb{Z})^*$ has precisely 2^r square roots. Hence given

$x \in \mathbb{Z}_n^*$, if $y \in \mathbb{Z}_n^*$ is randomly chosen such that $x^2 = y^2 \in (\mathbb{Z}/n\mathbb{Z})^*$, then with probability $\frac{2^r-2}{2^r} = 1 - \frac{1}{2^{r-1}}$ we have $x \neq \pm y \in (\mathbb{Z}/n\mathbb{Z})^*$. Hence if n is not a prime power this yields a factorisation of n with probability $\geq \frac{1}{2}$.

To find $x, y \in \mathbb{Z}_n^*$ such that $x^2 = y^2 \in (\mathbb{Z}/n\mathbb{Z})^*$ in the first place we proceed as follows: Let $B \in \mathbb{N}$ be a bound, and assume that all the primes $p_1, \ldots, p_t \in \mathbb{N}$ not exceeding B are known. Letting additionally $p_0 := -1$, the sequence $[p_0, p_1, \ldots, p_t]$ is called the associated **factor base**.

For $x \in \mathbb{Z}_n^*$ let $x' \in \{-\frac{n-1}{2}, \ldots, \frac{n-1}{2}\}$ such that $x^2 = x' \in (\mathbb{Z}/n\mathbb{Z})^*$. If $|x'| \in \mathbb{N}$ is B-smooth, then we have $x' = \prod_{i=0}^t p_i^{a_i(x^2)} \in \mathbb{Z}$, where $a_i(x^2) \in \mathbb{N}_0$ and $a_0(x^2) \leq 1$. In this case x is called a B-**number** and $a(x^2) := [a_0(x^2), \ldots, a_t(x^2)] \in \mathbb{N}_0^{t+1}$ is called the associated **exponent vector**; considering its entries as elements of $\mathbb{F}_2$ we get the **reduced** exponent vector $\overline{a}(x^2) := [\overline{a}_0(x^2), \ldots, \overline{a}_t(x^2)] \in \mathbb{F}_2^{t+1}$.

Let $x_1, \ldots, x_s \in \mathbb{Z}_n^*$ be B-numbers such that $\sum_{j=1}^s \overline{a}(x_j^2) = 0 \in \mathbb{F}_2^{t+1}$. Hence $\sum_{j=1}^s a_i(x_j^2) \in \mathbb{N}$ is even for $i \in \{0, \ldots, t\}$. Letting $x := \prod_{j=1}^s x_j \in \mathbb{Z}$ and $y := \prod_{i=0}^t p_i^{\frac{1}{2}\sum_{j=1}^s a_i(x_j^2)} \in \mathbb{Z}$ we have $y^2 = \prod_{i=0}^t p_i^{\sum_{j=1}^s a_i(x_j^2)} = \prod_{j=1}^s \prod_{i=0}^t p_i^{a_i(x_j^2)} = \prod_{j=1}^s x_j' \in \mathbb{Z}$, implying $y^2 = x^2 \in (\mathbb{Z}/n\mathbb{Z})^*$.

a) The random squares method [Dixon, 1981]. If the integers $x \in \mathbb{Z}_n^*$ are chosen randomly, this yields a Las-Vegas algorithm to factor n, and using the **Canfield-Erdős-Pomerance Theorem** [1983] estimating of the fraction of B-smooth integers in $\mathbb{Z}_n^*$ we get **Dixon's Theorem**: Letting $L(n) := L_{\frac{1}{2},1}(n) := e^{\sqrt{\ln(n)\ln\ln(n)}}$, we find a proper divisor of n using a factor base of size $t \sim \sqrt{L(n)}$ and $s \sim \sqrt{L(n)^3}$ tries with a probability $\geq 1 - 2e^{-\frac{1}{2}\sqrt{L(n)}}$ in subexponential time $O(L(n)^{2+o(1)})$.

In practice, the integers x are not chosen randomly but taken as $x \sim \lfloor\sqrt{n}\rfloor$, since then $|x^2 - n|$ is more likely to be B-smooth. We have to find at most $t + 2$ B-numbers until we arrive at a $\mathbb{F}_2$-linear dependency between the associated reduced exponent vectors, where the exponent vectors can be computed independently, and hence using distributed computing, while finding $\mathbb{F}_2$-linear dependencies has to be done using specially tailored Gaussian elimination techniques for sparse matrices over $\mathbb{F}_2$.

b) The continued fraction method [Morrison-Brillhart, 1971]. If $x^2 = z \in \mathbb{Z}/n\mathbb{Z}$ for some $z \in \mathbb{Z}$, then we have $x^2 = z + l^2kn$ for some $k, l \in \mathbb{Z}$. If $k, l > 0$ then $(\frac{x}{l})^2 - kn = \frac{z}{l^2}$ implies $|\frac{x}{l} - \sqrt{kn}| \leq \frac{1}{|\frac{x}{l}+\sqrt{kn}|} \cdot \frac{|z|}{l^2} \leq \frac{|z|}{l^2\sqrt{n}}$. Hence if $|z| < \frac{1}{2}\sqrt{n}$ we from Legendre's Theorem (9.3) get that $\frac{x}{l} \in \mathbb{Q}$ is a convergent of the continued fraction expansion of the quadratic irrationality $\sqrt{kn} \in \mathbb{R}$.

Hence for $k \in \mathbb{N}$ small we check for a few of its convergents $\rho_i = \frac{\sigma_i}{\tau_i} \in \mathbb{Q}$, where $i \in \mathbb{N}$ and $\sigma_i, \tau_i \in \mathbb{N}$ such that $\gcd(\sigma_i, \tau_i) = 1$, whether $x = \sigma_i \in \mathbb{N}$ is a B-number, i. e. letting $l = \tau_i$ whether $z = x^2 - l^2kn = \sigma_i^2 - \tau_i^2kn \in \mathbb{Z}$ is B-smooth. This yields a Las-Vegas algorithm to factor n, conjecturally having subexponential running time $O(L(n)^{\sqrt{\frac{3}{2}}+o(1)})$.

c) The quadratic sieve method (QS) [Pomerance, 1984]. Let $m := \lfloor\sqrt{n}\rfloor$ and $f := (X+m)^2 - n \in \mathbb{Z}[X]$. Hence it is likely that $f(x) \in \mathbb{Z}$ is B-smooth whenever $x \in \mathbb{Z}$ is small, thus yielding a whole bunch of B-numbers $x+m \in \mathbb{Z}$:

We choose a **sieve interval** $\{-C, \ldots, C\}$ for some $C \in \mathbb{N}$, assumed to be small compared to n. If $p \in \mathbb{N}$ is a prime, then from $f(x+p) = (x+m+p)^2 - n = f(x) + p(2x+2m+p)$ we conclude that $f(x) = 0 \in \mathbb{Z}/p\mathbb{Z}$ if and only if $f(x+kp) = 0 \in \mathbb{Z}/p\mathbb{Z}$ for all $k \in \mathbb{Z}$. If p is odd such that $\gcd(p,n) = 1$ and $f(x) = 0 \in \mathbb{Z}/p\mathbb{Z}$, then $n = (x+m)^2 \in (\mathbb{Z}/p\mathbb{Z})^*$ is a square, which by the cyclicity of $(\mathbb{Z}/p\mathbb{Z})^*$ is seen to be equivalent to $n^{\frac{p-1}{2}} = 1 \in (\mathbb{Z}/p\mathbb{Z})^*$; the latter is called **Euler's criterion**. Thus those p such that $n^{\frac{p-1}{2}} \neq 1 \in (\mathbb{Z}/p\mathbb{Z})^*$ can be discarded from the factor base. For the remaining p we first check whether $f(x) = 0 \in \mathbb{Z}/p\mathbb{Z}$, where $x \in \{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}$ whenever p is odd, and $x \in \{0,1\}$ for $p = 2$, and then compute the values of f at the other $x \in C$, increasing and decreasing x in steps of length p. Whenever we have $f(x) = 0 \in \mathbb{Z}/p\mathbb{Z}$, we divide out the maximum p-power dividing $f(x)$ and proceed with the quotient.

Since $m^2 - n$ is small and C is small compared to n, for $x \in \{-C, \ldots, C\}$ we have $f(x) = (x+m)^2 - n = x^2 + 2xm + (m^2 - n) \sim 2xm$, hence $|f(x)|$ is bounded by $\sim 2C\sqrt{n}$. The **Pomerance conjecture** says that the fraction of B-smooth integers amongst the $f(x)$, for $x \in \{-C, \ldots, C\}$, is asymptotically the same as the fraction of B-smooth integers in $\{1, \ldots, 2C\sqrt{n}\}$. Assuming the validity of this conjecture, this yields a Las-Vegas algorithm to factor n, and we have **Pomerance's Theorem**: We find a proper divisor of n with a probability $\geq 1 - 2e^{-\frac{1}{2}\sqrt[3]{L(n)}}$ in subexponential time $O(L(n)^{\sqrt{\frac{9}{8}}+o(1)})$.

In practice, for $n \sim 10^{50}$ factor bases of size $t \sim 3000$, by the Prime Number Theorem, see [5, Ch.22], corresponding to $B \sim 10^5$, and sieve intervals of size $C \sim 2 \cdot 10^5$ are used, while for $n \sim 10^{100}$ we choose $t \sim 50\,000$, corresponding to $B \sim 10^6$, and $C \sim 1.4 \cdot 10^7$; see [2, Tbl.9.1].

(17.5) Example. We apply the above quadratic form methods to $n := 7429$, hence $m := \lfloor\sqrt{n}\rfloor = 86$. The Fermat-Legendre method for $k := 1$ and $x := m+87 = 173$ yields $y = 150$, thus $x^2 - y^2 = 29929 - 22500 = n = (x-y)(x+y) = 23 \cdot 323$. We choose $B := 7$, hence the factor base is $[-1, 2, 3, 5, 7]$.

a) Using the random choice $x_1 := 6708$ and $x_2 := 2468$, yields the exponent vectors $a(x_1^2) = [1,0,3,0,1]$ and $a(x_2^2) = [1,2,3,0,1]$, hence $\frac{1}{2}(a(x_1^2) + a(x_2^2)) = [1,1,3,0,1]$, yielding $x = x_1x_2 = 16\,555\,344$ and $y = (-1)^1 \cdot 2^1 \cdot 3^3 \cdot 5^0 \cdot 7^1 = -378$, thus $\gcd(x+y, n) = 19$ and $\gcd(x-y, n) = 391$ and $n = 19 \cdot 391$.

b) The continued fraction method for $k := 1$ yields $\sqrt{n} = \mathrm{cf}[86, 5, 4, 1, 1, \ldots]$. We get the first convergent $\rho_1 = \frac{86}{1}$, thus $x = 86$ and $l = 1$, hence $z = x^2 - l^2 n = -33$, which is not B-smooth. The second convergent is $\rho_2 = \frac{431}{5}$, thus thus $x = 431$ and $l = 5$, hence $z = x^2 - l^2 n = 36$, yielding the exponent vector $a(x^2) = [0,2,2,0,0]$. Thus letting $y = (-1)^0 \cdot 2^1 \cdot 3^1 \cdot 5^0 \cdot 7^0 = 6$ yields $\gcd(x-y, n) = 17$ and $\gcd(x+y, n) = 437$ and $n = 17 \cdot 437$.

c) We choose $C := 3$ in the quadratic sieve method, hence the sieve interval is $\{-3, \ldots, 3\}$, and precisely $[-3, 1, 2]$ therein correspond to B-numbers:

x	-3	-2	-1	0	1	2	3
$(x+m)^2 - n$	-540	-373	-204	-33	140	315	492
sieve with 2	-135		-51		35		123
sieve with 3	-5		-17	-11		35	41
sieve with 5	-1				7	7	
sieve with 7					1	1	

The associated matrix of exponent vectors is $M := \begin{bmatrix} 1 & 2 & 3 & 1 & . \\ . & 2 & . & 1 & 1 \\ . & . & 2 & 1 & 1 \end{bmatrix} \in \mathbb{Z}^{3\times 5}$. Reduction yields $\begin{bmatrix} 1 & . & 1 & 1 & . \\ . & . & . & 1 & 1 \\ . & . & . & 1 & 1 \end{bmatrix} \in \mathbb{F}_2^{3\times 5}$, whose kernel is $\langle [0,1,1] \rangle_{\mathbb{F}_2}$. Letting $\frac{1}{2} \cdot [0,1,1] \cdot M = [0,1,1,1,1] \in \mathbb{Z}^5$ we get $y = (-1)^0 \cdot 2^1 \cdot 3^1 \cdot 5^1 \cdot 7^1 = 210$ and $x = (1+m) \cdot (2+m) = 87 \cdot 88 = 7656$, thus $\gcd(x-y, n) = 17$ and $\gcd(x+y, n) = 437$ and $n = 17 \cdot 437$.

(17.6) Other factorisation methods. In practice, trial division of composite $n \in \mathbb{N}$, running in exponential time $O(e^{(\frac{1}{2}+o(1))\ln(n)})$, is used to find prime divisors $p < 10^6$.

The $p-1$ method is based on the group $\mathbb{F}_p^* \cong C_{p-1}$, and is feasible whenever $p-1$ has only small prime divisors. Using the group $\mathbb{F}_{p^2}^*/\mathbb{F}_p^* \cong C_{p+1}$ instead yields the $p+1$ method [Guy, 1975; Williams, 1982], and even more generally there is the $\Phi_k(p)$ method [Bach-Shallit, 1988], where $\Phi_k \in \mathbb{Z}[X]$ is the k-th **cyclotomic polynomial**.

In the Elliptic Curve Method (ECM) [Lenstra, 1987], which also generalises the $p-1$ method, the group G of points of an elliptic curve over $\mathbb{F}_p$ is used instead. The **Hasse bound** yields $||G| - (p+1)| \leq 2\sqrt{p}$, hence again $|G| \sim p$, and varying the elliptic curve it is likely to find a group such that $|G|$ only has small prime divisors. Conjecturally it has Las-Vegas subexponential running time $O(L(n)^{1+o(1)})$; see [7, Ch.19] and [3, Ch.10.3].

Shanks's class group method [1969] and the Square Form Factorisation method (SQUFOF) [Shanks, 1972], exploiting the ideal class groups of imaginary and real quadratic number fields, respectively, have Las-Vegas exponential running time $O(e^{(\frac{1}{4}+o(1))\ln(n)})$. Combining these methods with the ideas of the $p-1$ method and the index calculus methods yields the Schnorr-Lenstra class group method [1984], which has Las-Vegas subexponential running time; see [3, Ch.8.6, 8.7, 10.2].

Generalising the quadratic sieve, the Multi-Polynomial Quadratic Sieve (MPQS) [Pomerance, 1987], conjecturally has Las-Vegas subexponential running time

$O(L(n)^{1+o(1)})$; see [3, Ch.10.4]. The Number Field Sieve (NFS) [Lenstra-Lenstra-Manasse-Pollard, 1990] generalises the idea from quadratic to general number fields, and conjecturally has Las-Vegas subexponential running time $O(L_{\frac{1}{3},c}(n))$, where $L_{\frac{1}{3},c}(n) := e^{c\sqrt[3]{\ln(n)}\sqrt[3]{(\ln\ln(n))^2}}$ and $c = \sqrt[3]{\frac{64}{9}}$; see [3, Ch.10.5].

IV Authentication

18 One-way functions

(18.1) One-way functions. Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} \subseteq \mathcal{X}^*$ and let $f\colon \mathcal{D} \to \mathcal{X}^*$ be in FP; hence in particular there is $k \in \mathbb{N}$ such that $l(f(w)) \leq l(w)^k$ for $w \in \mathcal{D}$. Letting $\mathcal{R} := \{[v,w] \in \mathrm{im}(f) \times \mathcal{D}; f(w) = v\} \subseteq \mathcal{X}^* \times \mathcal{X}^*$, assume that $l(w) \leq l(v)^k = l(f(w))^k$ for all $[v,w] \in \mathcal{R}$. Hence $\mathcal{R}$ is a polynomial certificate for the following decision problem: Given $v \in \mathcal{X}^*$, is $v \in \mathrm{im}(f)$? Thus the function problem associated to $\mathcal{R}$ is in FNP; if f is injective, then $\mathcal{R}$ represents the inverse $f^{-1}\colon \mathrm{im}(f) \to \mathcal{D}$ of f.

If the function problem associated to $\mathcal{R}$ is not in FP, then f is called a **one-way function**, if it even cannot be solved in expected Las-Vegas polynomial time, then f is called a **cryptographic one-way function**; since f is in FP Monte-Carlo and Las-Vegas algorithms for the function problem associated to $\mathcal{R}$ are polynomial time equivalent. If we had $\mathsf{FNP} = \mathsf{FP}$, which only by conjecture is not the case, there could not possibly exist any one-way functions at all. Still, candidates for one-way functions are the following:
a) Since integer multiplication needs polynomial time, but there are only Las-Vegas subexponential time algorithms for integer factorisation known, see (17.6), integer multiplication conjecturally is a cryptographic one-way function.
b) Since integer modular exponentiation needs polynomial time, but there are only Las-Vegas subexponential time algorithms for the discrete logarithm problem known, see [12, Ch.6.2] and [2, Ch.10] and [10, Ch.3.6], integer modular exponentiation conjecturally is a cryptographic one-way function.
c) Since integer modular squaring needs polynomial time, but taking square roots is Las-Vegas polynomial time equivalent to factoring the modulus, where the treatment of RSA moduli in (10.2) is straightforwardly generalised, modular squaring conjecturally is a cryptographic one-way function.
d) If the modulus is an odd prime p, taking modular square roots needs Las-Vegas polynomial time: Let $p-1 = 2^l m$, where $l \in \mathbb{N}$ and m is odd. By randomly choosing elements $z \in (\mathbb{Z}/p\mathbb{Z})^*$, and by Euler's criterion checking whether $z^{\frac{p-1}{2}} = -1$, we find a non-square in $(\mathbb{Z}/p\mathbb{Z})^*$, and thus $\alpha := z^m \in (\mathbb{Z}/p\mathbb{Z})^*$ has order 2^l. Letting $x \in (\mathbb{Z}/p\mathbb{Z})^*$ be a square, there is $k \in \mathbb{N}_0$ such $x^m = \alpha^{2k}$. Hence letting $y := x^{\frac{m+1}{2}}\alpha^{-k} \in (\mathbb{Z}/p\mathbb{Z})^*$ we have $y^2 = x^{m+1}\alpha^{-2k} = x \in (\mathbb{Z}/p\mathbb{Z})^*$, thus $\{\pm y\}$ are the square roots of $x \in (\mathbb{Z}/p\mathbb{Z})^*$.

If $p \equiv 3 \pmod 4$ or $p \equiv 5 \pmod 8$ taking modular square roots needs polynomial time by (10.1) and (26.12), respectively, but this in not known for the case $p \equiv 1 \pmod 8$. Hence integer modular squaring with prime modulus is not a cryptographic one-way function, but conjecturally is a one-way function.

(18.2) Unambiguous machines [Valiant, 1976]. A non-deterministic Turing machine $\mathcal{T}$ over an alphabet $\mathcal{X}$ is called **unambiguous**, if for all $w \in \mathcal{X}^*$ the machine $\mathcal{T}$ halts, and there is at most one accepting branch. The complexity class UP is the set of languages being non-deterministically decidable by an unambiguous non-deterministic Turing machine running in polynomial time.

We have $\mathsf{P} \subseteq \mathsf{UP} \subseteq \mathsf{NP}$, where it is conjectured that $\mathsf{P} \neq \mathsf{UP}$ and $\mathsf{UP} \neq \mathsf{NP}$.

(18.3) Theorem. $\mathsf{P} \neq \mathsf{UP}$ if and only if there is an injective one-way function.

Proof. Let $f \colon \mathcal{D} \rightarrow \mathcal{X}^*$ be an injective one-way function, where $\mathcal{X}$ is an alphabet and $\mathcal{D} \subseteq \mathcal{X}^*$. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be totally ordered by $x_1 < x_2 < \cdots < x_n$, and let $\mathcal{X}^*$ be totally ordered length-lexicographically, the order also being denoted by $\leq$. Let $\mathcal{L}_f := \{[w, v] \subseteq \mathcal{X}^* \times \mathcal{X}^* \times ; v = f(u) \text{ for some } u \in \mathcal{D}, u \leq w\}$. We show that $\mathcal{L}_f \in \mathsf{UP} \setminus \mathsf{P}$:

Let $\mathcal{T}$ be the following non-deterministic Turing machine: Since f is a one-way function, there is $k \in \mathbb{N}$ such that for all $u, v \in \mathcal{X}^*$ such that $f(u) = v$ we have $l(u) \leq l(v)^k$. Hence given $[w, v]$, the machine $\mathcal{T}$ chooses $u \in \mathcal{X}^*$ such that $l(u) \leq l(v)^k$, and checks whether $f(u) = v$ holds. If this is the case, it checks whether $u \leq w$ holds, and in this case the corresponding branch accepts, otherwise the branch rejects. Since f is in FP, the machine $\mathcal{T}$ decides $\mathcal{L}_f$ in polynomial time, and since f is injective, $\mathcal{T}$ is unambiguous, thus $\mathcal{L}_f \in \mathsf{UP}$.

Assume that $\mathcal{L}_f$ is in P. Then for $v \in \mathcal{X}^*$ we find $f^{-1}(v)$ by n-ary search as follows: If $[x_n^{l(v)^k}, v] \notin \mathcal{L}_f$, then $v \notin \operatorname{im}(f)$. Otherwise, we determine the largest $i \in \mathbb{N}_0$ such that $[x_n^{l(v)^k - i}, v] \in \mathcal{L}_f$, hence we have $v = f(u)$ where $u \in \mathcal{X}^*$ such that $l := l(u) = l(v)^k - i$. We determine the initial letter $x_j \in \mathcal{X}$ of u by finding the smallest $j \in \mathbb{N}$ such that $[x_j x_n^{l-1}, v] \in \mathcal{L}_f$. Keeping j fixed, we proceed by considering $[x_j x_{j'} x_n^{l-2}, v]$ for $j' \in \mathbb{N}$, and so on. Thus u can be computed by at most $i + nl \leq (1 + |\mathcal{X}|) \cdot l(v)^k$ runs of the polynomial time Turing machine deciding $\mathcal{L}_f$, hence f^{-1} is in FP, a contradiction.

Conversely, let $\mathcal{L}$ be in $\mathsf{UP} \setminus \mathsf{P}$, and let $\mathcal{T}$ be a polynomial time unambiguous non-deterministic Turing machine over $\mathcal{X}$ deciding $\mathcal{L}$. We define $f_{\mathcal{T}} \colon \mathcal{D} \rightarrow \mathcal{X}^*$ as follows: Let $\mathcal{D} \subseteq \mathcal{X}^*$ be the set of all $v_w \in \mathcal{X}^*$ encoding an accepting computation for some $w \in \mathcal{X}^*$, and let $f_{\mathcal{T}}(v_w) := w$. We show that $f_{\mathcal{T}}$ is an injective one-way function: Since the input w of a computation can be read off from the encoding v_w the function $f_{\mathcal{T}}$ is well-defined and in FP, since $\mathcal{T}$ is unambiguous $f_{\mathcal{T}}$ is injective, and since $\mathcal{T}$ runs in polynomial time we have $l(v_w) \leq l(w)^k$ for some $k \in \mathbb{N}$. Assume that $f_{\mathcal{T}}^{-1}$ is in FP, then $f_{\mathcal{T}}^{-1}(w)$ for $w \in \mathcal{X}^*$ decides in polynomial time whether $w \in \mathcal{L}$, a contradiction. ♯

(18.4) Trapdoor functions. Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} \subseteq \mathcal{X}^*$ and let $f\colon \mathcal{D} \to \mathcal{X}^*$ be an injective one-way function. Let moreover $t\colon \mathcal{D} \to \mathcal{X}^*$ be in FP, such that there is a function $f_t\colon \mathrm{im}(f) \times \mathrm{im}(t) \to \mathcal{D}\colon [f(w), t(w)] \mapsto w$ which is in FP. Then t is called a **(deterministic) trapdoor function** for f. If f_t can be computed in expected Las-Vegas polynomial time, then t is called a **cryptographic trapdoor function**.

E. g. we consider the RSA function $f\colon [p, q, e, x] \to [pq, e, x^e]$, where $p \neq q \in \mathbb{N}$ are odd primes, $e \in (\mathbb{Z}/\varphi(pq)\mathbb{Z})^*$ and $x \in (\mathbb{Z}/pq\mathbb{Z})^*$. Hence f is an injective function in FP, and since inverting f involves integer factorisation, f is a one-way function if integer multiplication is. Letting $d := e^{-1} \in (\mathbb{Z}/\varphi(pq)\mathbb{Z})^*$, the component x of $[p, q, e, x]$ can be computed in polynomial time from d and $[pq, e, x^e]$, hence $t\colon [p, q, e, x] \to d$ is a trapdoor function for the component x. The modulus pq can be factored using e and d in Las-Vegas polynomial time, see (9.2), hence t is a cryptographic trapdoor function for p and q.

19 Hash functions

(19.1) Hash functions. a) Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} \subseteq \mathcal{X}^*$ and let $n \in \mathbb{N}$. A function $h\colon \mathcal{D} \to \mathcal{X}^{\leq n}$ in FP is called a **hash function**; if additionally $\mathcal{D}$ is finite such that $|\mathcal{D}| \geq |\mathcal{X}^{\leq n}|$ then h is called a **compression function**. The elements of $\mathcal{D} \subseteq \mathcal{X}^*$ are called the **messages**, and the **hash values** $h(w)$ for $w \in \mathcal{D}$ are also called **message digests** or **authentication tags**.

A function $h\colon \mathcal{H} \times \mathcal{D} \to \mathcal{X}^{\leq n}\colon [u, w] \mapsto h_u(w)$ in FP, where $\mathcal{H} \subseteq \mathcal{X}^*$ is called the set of **keys**, is called a **keyed hash function**; if additionally $\mathcal{D}$ is finite such that $|\mathcal{D}| \geq |\mathcal{X}^{\leq n}|$ then h is called a **keyed compression function**.

Hash functions are used for **modification detection codes (MDCs)**: To assure that some message $w \in \mathcal{D}$, which is to be made public, is not changed afterwards, its hash value $h(w) \in \mathcal{X}^{\leq n}$ is stored privately, and later on for the possibly modified message $w' \in \mathcal{X}^*$ it is checked whether still $w' \in \mathcal{D}$ and $h(w') = h(w)$.

Keyed hash functions are used for **message authentication codes (MACs)**: If Alice and Bob agree on a private key $u \in \mathcal{H}$, then Bob publicly sends the message-hash value pair $[w, h_u(w)] \in \mathcal{D} \times \mathcal{X}^{\leq n}$ to Alice, and Alice verifies whether the received pair $[w', v'] \in \mathcal{X}^* \times \mathcal{X}^*$ fulfils $w' \in \mathcal{D}$ and $h_u(w') = v'$.

b) Associated with a hash function $h\colon \mathcal{D} \to \mathcal{X}^{\leq n}$ we have the following functions problems not necessarily in FNP: If the function problem associated to $\{[b, v; w] \in \mathbb{N}_0 \times \mathcal{X}^{\leq n} \times \mathcal{D}; l(w) = b, h(w) = v\}$, i. e. given $b \in \mathbb{N}_0$ and $v \in \mathcal{X}^{\leq n}$, find $w \in \mathcal{D}$ such that $l(w) = b$ and $h(w) = v$, cannot be solved in expected Monte-Carlo polynomial time, the input length being in $O(n + \ln b)$, then h is called **preimage resistant**.

If the function problem associated to $\{[w; w'] \in \mathcal{D} \times \mathcal{D}; w \neq w', h(w) = h(w')\}$, i. e. given $w \in \mathcal{D}$, find a **collision** $w \neq w' \in \mathcal{D}$ such that $h(w) = h(w')$, cannot be solved in expected Monte-Carlo polynomial time, the input length being $l(w)$, then h is called **second-preimage resistant** or **weakly collision resistant**.

If the function problem associated to $\{[b; w, w'] \in \mathbb{N}_0 \times \mathcal{D} \times \mathcal{D}; l(w) = b, l(w') \leq b, w \neq w', h(w) = h(w')\}$, i. e. given $b \in \mathbb{N}_0$, find $w \neq w' \in \mathcal{D}$ such that $l(w) = b$ and $l(w') \leq b$ and $h(w) = h(w')$, cannot be solved in expected Monte-Carlo polynomial time, the input length being in $O(\ln b)$, then h is called **(strongly) collision resistant**.

(19.2) Random oracle model [Bellare-Rogaway, 1993]. We assume the compression function $h \colon \mathcal{D} \to \mathcal{X}^{\leq n}$ to be chosen randomly from the set $(\mathcal{X}^{\leq n})^{\mathcal{D}}$ of all functions from $\mathcal{D}$ to $\mathcal{X}^{\leq n}$, where we assume the uniform probability distribution on $(\mathcal{X}^{\leq n})^{\mathcal{D}}$. Hash values are computed by an oracle, and are independent in the following sense: Let $\mathcal{D}' \subseteq \mathcal{D}$ and $v_{w'} \in \mathcal{X}^{\leq n}$ for all $w' \in \mathcal{D}'$ be given. Then for $w \in \mathcal{D} \setminus \mathcal{D}'$ and $v \in \mathcal{X}^{\leq n}$ we have the conditional probability $\mu(h(w) = v | h(w') = v_{w'}$ for all $w' \in \mathcal{D}') = \frac{1}{t}$, where $t := |\mathcal{X}^{\leq n}| \in \mathbb{N}$; we have $t = |\mathcal{X}^{\leq n}| = \sum_{i=0}^{n} |\mathcal{X}|^i = \frac{|\mathcal{X}|^{n+1}-1}{|\mathcal{X}|-1}$ if $|\mathcal{X}| > 1$, and $t = n+1$ if $|\mathcal{X}| = 1$.

a) To find a collision we query the oracle for $\mathcal{D}' \subseteq \mathcal{D}$ where $k := |\mathcal{D}'| \in \{1, \ldots, t\}$, and for $w \in \mathcal{D}'$ store the hash value $h(w)$ and compare it with the hash values found earlier. Hence the success probability is $\epsilon = 1 - \prod_{i=0}^{k-1}(1 - \frac{i}{t})$, which using $e^{-\lambda} \geq 1 - \lambda \geq 0$ for $0 \leq \lambda \leq 1$, where $|e^{-\lambda} - (1-\lambda)| \leq \frac{\lambda^2}{2}$, yields $\epsilon \geq 1 - \prod_{i=0}^{k-1} e^{-\frac{i}{t}} = 1 - e^{\frac{-k(k-1)}{2t}}$. Thus we obtain $k(k-1) \geq -2t\ln(1-\epsilon)$, hence we arrive again at the birthday paradox $k \geq \sqrt{-2t\ln(1-\epsilon)}$. Thus given a fixed success probability $0 < \epsilon < 1$ we need $k \in O(\sqrt{t})$ queries to find a collision; in practice, where $|\mathcal{X}| = 2$, to avoid the **birthday attack** or **square root attack** $n \sim 128$ is recommended, yielding $t = 2^{129} - 1$ and $k \sim 2^{64}$.

b) Given $w \in \mathcal{D}$, to find a second preimage $w \neq w' \in \mathcal{D}$ such that $h(w) = h(w')$, we query the oracle for $w \notin \mathcal{D}' \subseteq \mathcal{D}$ where $|\mathcal{D}'| = k$, hence the success probability is $\epsilon = 1 - (1 - \frac{1}{t})^k$.

Anyway, not necessarily assuming the random oracle model, finding a collision reduces to finding a second preimage, by choosing $w \in \mathcal{D}$ and computing a second preimage $w \neq w' \in \mathcal{D}$ of $h(w)$. If a second preimage is found with success probability ϵ, then a collision is also found with success probability ϵ.

c) Given $v \in \mathcal{X}^{\leq n}$, to find a preimage $w \in \mathcal{D}$ we query the oracle for $\mathcal{D}' \subseteq \mathcal{D}$ where $|\mathcal{D}'| = k$, hence the success probability is $\epsilon = 1 - (1 - \frac{1}{t})^k$.

Anyway, not necessarily assuming the random oracle model, finding a collision reduces to finding a preimage, by choosing $w \in \mathcal{D}$, computing a preimage $w' \in \mathcal{D}$ of $h(w)$, and checking whether $w \neq w'$. If a preimage is found with success probability ϵ, yielding any preimage of $h(w)$ with the same probability, then a collision is found with success probability $\epsilon(1 - \frac{1}{|h^{-1}(h(w))|})$ for fixed $w \in \mathcal{D}$, and averaging over $\mathcal{D}$ yields a success probability $\frac{\epsilon}{|\mathcal{D}|} \cdot \sum_{w \in \mathcal{D}}(1 - \frac{1}{|h^{-1}(h(w))|}) = \frac{\epsilon}{|\mathcal{D}|} \cdot (|\mathcal{D}| - \sum_{v \in \mathrm{im}(h)} \frac{|h^{-1}(v)|}{|h^{-1}(v)|}) = \epsilon(1 - \frac{|\mathrm{im}(h)|}{|\mathcal{D}|})$.

(19.3) Chaum-van Heijst-Pfitzmann function [1992]. Let $q \in \mathbb{N}$ be an odd prime such that $p := 2q+1 \in \mathbb{N}$ is a prime as well, being called a pair of **Germain primes**; it is an open problem whether there are infinitely many such pairs, but it is conjectured that $|\{q \in \mathbb{N}; q \leq n, q, 2q+1 \text{ prime}\}| \sim \frac{2cn}{\ln(n)^2}$ for some $c > 0$ [Hardy-Littlewood, 1922].

Let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root, let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$, and let $h_{\rho,\sigma} \colon \mathbb{Z}_q \times \mathbb{Z}_q \to (\mathbb{Z}/p\mathbb{Z})^* \colon [x,y] \mapsto \rho^x\sigma^y$; hence $h_{\rho,\sigma}$ is in FP, and since $q^2 > p-1$ this is a compression function. We show that finding the discrete logarithm $b := \log_\rho(\sigma) \in \mathbb{Z}/(p-1)\mathbb{Z}$ reduces to finding collisions of $h_{\rho,\sigma}$:

Let $[x,y] \neq [x',y'] \in \mathbb{Z}_q \times \mathbb{Z}_q$ such that $\rho^x\sigma^y = h_{\rho,\sigma}(x,y) = h_{\rho,\sigma}(x',y') = \rho^{x'}\sigma^{y'} \in (\mathbb{Z}/p\mathbb{Z})^*$. Assume that $y = y'$, then we conclude $\rho^x = \rho^{x'}$ and thus $2q = p-1 \mid x-x'$, and since $|x-x'| < q$ we have $x = x'$, a contradiction. Hence we have $y \neq y'$. We consider the equation $x - x' = z(y'-y) \in \mathbb{Z}/(p-1)\mathbb{Z}$ for $z \in \mathbb{Z}/(p-1)\mathbb{Z}$. Since $\rho^{x-x'} = \sigma^{y'-y} = \rho^{b(y'-y)}$ shows that this equation at least has the solution $z = b$, we conclude that $g := \gcd(2q, y'-y) = \gcd(p-1, y'-y) \mid (x-x')$, where $|y-y'| < q$ implies that $g \in \{1,2\}$. If $g = 1$ then $y'-y \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, and hence we have the unique solution $b = \frac{x-x'}{y'-y} \in \mathbb{Z}/(p-1)\mathbb{Z}$. If $g = 2$ then $\gcd(q, y'-y) = 1$, and thus the equation $x - x' = z(y'-y) \in \mathbb{Z}/q\mathbb{Z}$ has the unique solution $c := \frac{x-x'}{y'-y} \in \mathbb{Z}/qZ$, implying that $b \in \{c, c+q\}$. ♯

(19.4) Merkle-Damgard construction [1990]. a) Let $\mathcal{X} = \{0,1\}$, and let $h \colon \mathcal{X}^{n+t} \to \mathcal{X}^n$ be a compression function, where $n, t \in \mathbb{N}$ such that $t \geq 2$. We construct a (keyed) **iterated hash function** $\widehat{h} \colon \mathcal{X}^n \times \mathcal{X}^{\geq n+t+1} \to \mathcal{X}^n$, using a **padding function** $g \colon \mathcal{X}^{\geq n+t+1} \to \coprod_{j \geq 2} \mathcal{X}^{j(t-1)}$ and an iteration process:

Padding is done as follows: Let $w \in \mathcal{X}^{\geq n+t+1}$. Then $w = w_1w_2\cdots w_kw_{k+1}$, where $k = \lfloor \frac{l(w)}{t-1} \rfloor$, and $l(w_i) = t-1$ for $i \in \{1,\ldots,k\}$, and $l(w_{k+1}) = t-1-d$ where $d \in \mathbb{Z}_{t-1}$. Let $g \colon \mathcal{X}^{\geq n+t+1} \to \coprod_{j \geq 2} \mathcal{X}^{j(t-1)}$ be defined by $g(w) := w'_1w'_2\cdots w'_kw'_{k+1}w'_{k+2}$, where $w'_i = w_i$ for $i \in \{1,\ldots,k\}$ and $w'_{k+1} := w_{k+1}0^d$ and $w'_{k+2} = 0^{t-1-b_2(d)}\beta(d)$, where $\beta(d) \in \mathcal{X}^{b_2(d)}$ is the binary representation of $d \in \mathbb{N}$. Since $t \geq 2$ we have $b_2(d) = 1 + \lfloor \log_2(d) \rfloor \leq 1 + \lfloor \log_2(t-1) \rfloor \leq t-1$, and hence we have $l(w'_i) = t-1$ for $i \in \{1,\ldots,k+2\}$; the function g is in FP and is injective, hence does not have collisions.

Iteration is done as follows: Let $u \in \mathcal{X}^n$ be a key; for the use as an unkeyed hash function we may let $u := 0^n \in \mathcal{X}^n$. For $w' = w'_1w'_2\cdots w'_j \in \mathcal{X}^{j(t-1)}$ such that $l(w'_i) = t-1$ for $i \in \{1,\ldots,j\}$, let $v_1,\ldots,v_j \in \mathcal{X}^n$ be defined successively by $v_1 := h(u0w'_1)$ and $v_i := h(v_{i-1}1w'_i)$ for $i \in \{2,\ldots,j\}$. Let $\widehat{h}_u \colon \mathcal{X}^{\geq n+t+1} \to \mathcal{X}^n$ be defined as $\widehat{h}_u(w) := v_{\lfloor \frac{l(w)}{t-1} \rfloor + 2}(g(w))$; to compute $\widehat{h}_u(w)$ for $w \in \mathcal{X}^{\geq n+t+1}$ we need $\lfloor \frac{l(w)}{t-1} \rfloor + 2$ queries of h, hence $\widehat{h}$ is in FP.

b) We show that finding a collision of h reduces to finding a collision of $\widehat{h}$: Assume we have a collision $w \neq \widetilde{w} \in \mathcal{X}^{\geq n+t+1}$ of $\widehat{h}_u$, then we need $O(l(w)+l(\widetilde{w}))$ queries of h to find a collision of h; to this we distinguish three cases:

Firstly, let $l(w) \not\equiv l(\widetilde{w}) \pmod{t-1}$, hence $d \neq \widetilde{d}$ and $w'_{k+2} \neq \widetilde{w}'_{\widetilde{k}+2}$. Since $h(v_{k+1}1w'_{k+2}) = v_{k+2} = \widetilde{v}_{\widetilde{k}+2} = h(\widetilde{v}_{\widetilde{k}+1}1\widetilde{w}'_{\widetilde{k}+2})$ we have found a collision of h.

Secondly, let $l(w) = l(\widetilde{w})$. Hence we have $k = \widetilde{k}$ and $w'_{k+2} = \widetilde{w}'_{k+2}$, thus $h(v_{k+1}1w'_{k+2}) = v_{k+2} = \widetilde{v}_{k+2} = h(\widetilde{v}_{k+1}1\widetilde{w}'_{k+2})$, and either we have found a collision of h, or we have $v_{k+1} = \widetilde{v}_{k+1}$. In the latter case we in turn have $h(v_k1w'_{k+1}) = v_{k+1} = \widetilde{v}_{k+1} = h(\widetilde{v}_k1\widetilde{w}'_{k+1})$, and hence proceeding backwards we find a collision of h, since otherwise we finally get $w'_i = \widetilde{w}'_i$ for $i \in \{1, \ldots, k+2\}$, implying $g(w) = g(\widetilde{w})$, a contradiction.

Thirdly, let $l(w) \equiv l(\widetilde{w}) \pmod{t-1}$ and $l(w) < l(\widetilde{w})$. Hence we have $k < \widetilde{k}$, and proceeding as above, we either find a collision of h, or we finally get $h(u0w'_1) = v_1 = \widetilde{v}_{\widetilde{k}-k+1} = h(\widetilde{v}_{\widetilde{k}-k}1\widetilde{w}'_{\widetilde{k}-k+1})$, where since $l(\widetilde{v}_{\widetilde{k}-k}) = n = l(u)$, the arguments of h differ at their $(n+1)$-st entry, hence we have found a collision of h. ♯

(19.5) Merkle-Damgard construction for $t = 1$. a) Let $\mathcal{X} = \{0, 1\}$ and let $h\colon \mathcal{X}^{n+1} \to \mathcal{X}^n$ be a compression function, where $n \in \mathbb{N}$. Similarly, we construct $\widehat{h}\colon \mathcal{X}^n \times \mathcal{X}^{\geq n+2} \to \mathcal{X}^n$ as follows:

Padding is done as follows: Let $w = x_1x_2\cdots x_k \in \mathcal{X}^{\geq n+2}$, where $x_i \in \mathcal{X}$ and $k := l(w)$, let $g'\colon \mathcal{X} \to \mathcal{X}^{\leq 2}$ be defined by $g'(0) = 0$ and $g'(1) = 01$, and let $g\colon \mathcal{X}^{\geq n+2} \to \mathcal{X}^{\geq n+4}$ be defined by $g(w) := 11 \cdot g'(x_1)\cdots g'(x_k)$. Hence g is in FP, and is injective. Except of the **prefix** 11, the word $g(w)$ does not contain the subword 11, hence whenever $w, \widetilde{w} \in \mathcal{X}^{\geq n+2}$ such that $g(w) = vg(\widetilde{w})$ for some $v \in \mathcal{X}^*$ we have $v = \epsilon$ and thus $w = \widetilde{w}$, implying that $g(w)$ is not a **postfix** of $g(\widetilde{w})$ where $w \neq \widetilde{w}$.

Iteration is done as follows: Let $u \in \mathcal{X}^n$ be a key; for the use as an unkeyed hash function we may let $u := 0^n$. For $w' = x'_1x'_2\cdots x'_j \in \mathcal{X}^j$ let $v_1, \ldots, v_j \in \mathcal{X}^n$ be defined successively by $v_1 := h(ux'_1)$ and $v_i := h(v_{i-1}x'_i)$ for $i \in \{2, \ldots, j\}$. Let $\widehat{h}_u\colon \mathcal{X}^{\geq n+2} \to \mathcal{X}^n$ be defined as $\widehat{h}_u(w) := v_{l(g(w))}(g(w))$; to compute $\widehat{h}_u(w)$ for $w \in \mathcal{X}^{\geq n+2}$ we need $l(g(w)) \leq 2l(w) + 2$ queries of h, hence $\widehat{h}$ is in FP.

b) We show that finding a collision of h reduces to finding a collision of $\widehat{h}$: Assume we have a collision $w \neq \widetilde{w} \in \mathcal{X}^{\geq n+2}$ of $\widehat{h}_u$, then we need $O(l(w) + l(\widetilde{w}))$ queries of h to find a collision of h; to this end we distinguish two cases:

Firstly, let $j := l(w') = l(\widetilde{w}')$. Thus we have $h(v_{j-1}x'_j) = v_j = \widetilde{v}_j = h(\widetilde{v}_{j-1}\widetilde{x}'_j)$, and either we have found a collision of h, or we have $v_{j-1} = \widetilde{v}_{j-1}$ and $x'_j = \widetilde{x}'_j$. In the latter case, proceeding backwards we find a collision of h, since otherwise we finally get $x'_i = \widetilde{x}'_i$ for $i \in \{1, \ldots, j\}$, implying $g(w) = g(\widetilde{w})$, a contradiction.

Secondly, we may assume that $j := l(w') < l(\widetilde{w}') =: \widetilde{j}$. Proceeding as above, we either find a collision of h, or we finally get $h(ux'_1) = v_1 = \widetilde{v}_{\widetilde{j}-j+1} = h(\widetilde{v}_{\widetilde{j}-j}\widetilde{x}'_{\widetilde{j}-j+1})$, where $x'_i = \widetilde{x}'_{\widetilde{j}-j+i}$ for $i \in \{1, \ldots, j\}$, implying that $g(\widetilde{w}) = vg(w)$ for some $v \in \mathcal{X}^{\widetilde{j}-j}$, a contradiction. ♯

(19.6) Remark. In practice, the **Secure Hash Algorithm** SHA-1 [1995] is the current standard. It follows the idea of iterated hash functions, and is based on a compression function $\mathcal{X}^{n+t} \to \mathcal{X}^n$, where $\mathcal{X} := \{0,1\}$ and $n = 160$ and $t = 512$, and where the particular padding function used maps $\mathcal{X}^{\leq m} \to \mathcal{X}^t$, where $m = 2^{64} - 1$. No collisions of SHA-1 are currently known; for further comments, on predecessors and successful attacks against those, as well as on proposed new standards, see [12, Ch.4.3.2] and [10, Ch.9.4].

20 Message authentication

(20.1) Forgeries. Let $\mathcal{X}$ be an alphabet, let $\mathcal{D}, \mathcal{H} \subseteq \mathcal{X}^*$, and let $h\colon \mathcal{H} \times \mathcal{D} \to \mathcal{X}^{\leq n}$, where $n \in \mathbb{N}$, be a keyed hash function. A pair $[w, w'] \in \mathcal{X}^* \times \mathcal{X}^*$ such that $w \in \mathcal{D}$ and $h_u(w) = w'$ is called **valid** for h with respect to u. A valid pair $[w, w'] \in \mathcal{X}^* \times \mathcal{X}^*$ is called an **(existential) forgery**, if it has been found without querying $h_u(w)$; queries $h_u(\widetilde{w})$ for $w \neq \widetilde{w} \in \mathcal{D}$ are allowed. A forgery $[w, w'] \in \mathcal{X}^* \times \mathcal{X}^*$ with prescribed message $w \in \mathcal{D}$ is called a **selective forgery**.

(20.2) Remark. A simple idea to construct a MAC is to incorporate a private key into an iterated hash function. However, this has to be done carefully:

Let $\mathcal{X} := \{0,1\}$ and let $h\colon \mathcal{X}^{n+t} \to \mathcal{X}^n$ be a compression function, where $n, t \in \mathbb{N}$. Let $g\colon \mathcal{X}^* \to \coprod_{j \geq 1} \mathcal{X}^{jt}\colon w \mapsto w\widehat{w}$ be a padding function in FP. Let $u \in \mathcal{X}^n$ be a key. For $w' = w'_1 w'_2 \cdots w'_j \in \mathcal{X}^{jt}$, where $l(w'_i) = t$ for $i \in \{1, \ldots, j\}$, let $v_1, \ldots, v_j \in \mathcal{X}^n$ be defined successively by $v_1 := h(uw'_1)$ and $v_i := h(v_{i-1}w'_i)$ for $i \in \{2, \ldots, j\}$. Let $\widehat{h}\colon \mathcal{X}^n \times \mathcal{X}^* \to \mathcal{X}^n$ be defined as $\widehat{h}_u(w) := v_{l(g(w))}(g(w))$; to compute $\widehat{h}_u(w)$ for $w \in \mathcal{X}^*$ we need $l(g(w))$ queries of h, hence $\widehat{h}$ is in FP.

To find a forgery, we proceed as follows: Let a valid pair $[w, \widehat{h}_u(w)] \in \mathcal{X}^* \times \mathcal{X}^n$ be known, and let $\widetilde{w} \in \mathcal{X}^*$. Letting $j := \frac{l(g(w))}{t} \geq 1$ we have $v_j(g(g(w)\widetilde{w})) = v_j(w\widehat{w}\widetilde{w}\widehat{\widetilde{w}}) = v_j(w\widehat{w}) = v_j(g(w)) = \widehat{h}_u(w)$. Thus $\widehat{h}_u(g(w)\widetilde{w}) = v_{\widetilde{j}}(g(g(w)\widetilde{w}))$, where $\widetilde{j} := \frac{l(g(g(w)\widetilde{w}))}{t} \geq j$, can be computed without knowing the key $u \in \mathcal{X}^n$.

(20.3) CBC-MACs [1985; Bellare-Kilian-Rogaway, 2000]. The **cipher block chaining (CBC)** mode of operation for block ciphers is used to define a keyed hash function: Let $\mathcal{X} := \{0,1\}$, let $\mathcal{C} := \mathcal{X}^m$ where $m \in \mathbb{N}$, and let $[\mathcal{C}, \mathcal{C}, \mathcal{H}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem where $\mathcal{H} \subseteq \mathcal{X}^*$ and where we assume that $E_u \in \mathcal{E}$ is in FP for $u \in \mathcal{H}$. For $n \in \mathbb{N}$ let $h\colon \mathcal{H} \times \mathcal{C}^n \to \mathcal{C}$ be the keyed hash function defined as follows: For $w = w_1 \cdots w_n \in \mathcal{C}^n$ let $v_0, \ldots, v_n \in \mathcal{C}$ be successively defined by $v_0 := 0^m \in \mathcal{C}$ and $v_i := E_u(v_{i-1} \oplus w_i) \in \mathcal{C}$ for $i \in \{1, \ldots, n\}$, where $\oplus$ denotes the 'exclusive or' operation on bit strings, then let $h_u(w) := v_n$; since E_u is in FP so is h.

Assuming appropriate security properties of the underlying cryptosystem, CBC-MACs are considered to be secure. The best known attack is a birthday chosen-message attack: Let $n \geq 2$, let $k \in \mathbb{N}$, let $w_0 \in \mathcal{C}$, let $w_{1,1}, \ldots, w_{1,k} \in \mathcal{C}$ be pairwise distinct, let $w_{2,1}, \ldots, w_{2,k} \in \mathcal{C}$, let $w_3, \ldots, w_n \in \mathcal{C}$, and let $\widetilde{w}_i :=$

$w_{1,i}w_{2,i}w_3\cdots w_n \in \mathcal{C}^n$ for $i \in \{1,\ldots,k\}$. Since $w_{1,i} \neq w_{1,j}$ we have $\widetilde{w}_i \neq \widetilde{w}_j$ for all $i \neq j$. Querying $h_u(\widetilde{w}_i)$ for $i \in \{1,\ldots,k\}$, by way of birthday attack assume that for some $i \neq j \in \{1,\ldots,k\}$ we have $h_u(\widetilde{w}_i) = h_u(\widetilde{w}_j)$. Since $E_u\colon \mathcal{C} \to \mathcal{C}$ is bijective, this is equivalent to $v_2(w_{1,i}w_{2,i}) = v_2(\widetilde{w}_i) = v_2(\widetilde{w}_j) = v_2(w_{1,j}w_{2,j})$, which in turn is equivalent to $v_1(w_{1,i}) \oplus w_{2,i} = v_1(w_{1,j}) \oplus w_{2,j}$. Letting $w := w_{1,i}(w_{2,i} \oplus w_0)w_3\cdots w_n \in \mathcal{C}^n$ and $\widetilde{w} := w_{1,j}(w_{2,j} \oplus w_0)w_3\cdots w_n \in \mathcal{C}^n$. Hence we have $w \neq \widetilde{w}$ and $h_u(w) = h_u(\widetilde{w})$, and querying $h_u(w)$ yields the forgery $[\widetilde{w}, h_u(w)] \in \mathcal{C}^n \times \mathcal{C}$; by the birthday paradox we obtain a forgery with success probability $0 < \epsilon < 1$ using $1 + \sqrt{-2^{m+1}\ln(1-\epsilon)} \in O(2^{\frac{m}{2}})$ queries of h.

(20.4) Nested MACs. a) Let $\mathcal{X}$ be an alphabet, let $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{X}^*$ and $\mathcal{G}, \mathcal{H} \subseteq \mathcal{X}^*$, and let $g\colon \mathcal{G} \times \mathcal{D} \to \mathcal{X}^{\leq m}$ and $h\colon \mathcal{H} \times \mathcal{D}' \to \mathcal{X}^{\leq n}$, where $m, n \in \mathbb{N}$, be keyed hash functions such that $\mathrm{im}(g) \subseteq \mathcal{D}'$. The composition $gh\colon (\mathcal{H} \times \mathcal{G}) \times \mathcal{D} \to \mathcal{X}^{\leq n}\colon [v, u; w] \mapsto h_v(g_u(w))$ is called a **nested** keyed hash function.

We show that finding a collision of g_u or a forgery for h_v reduces to finding a forgery for g_uh_v: Let $u \in \mathcal{G}$ and $v \in \mathcal{H}$, and let $[w, w''] \in \mathcal{D} \times \mathcal{X}^{\leq n}$ be a forgery for g_uh_v, found using $k \in \mathbb{N}$ queries of both g_u and h_v. Hence for some $w_1, \ldots, w_k \in \mathcal{D}$, where $w \neq w_i$ for $i \in \{1,\ldots,k\}$, we have found $w_i' := g_u(w_i) \in \mathcal{X}^{\leq m}$ and $w_i'' := h_v(w_i') \in \mathcal{X}^{\leq n}$. By another query of g_u let $w' := g_u(w) \in \mathcal{X}^{\leq m}$. If $w' = w_i'$ for some $i \in \{1,\ldots,k\}$, then $w \neq w_i \in \mathcal{D}$ is a collision of g_u. If $w' \neq w_i'$ for all $i \in \{1,\ldots,k\}$, then $[w', w''] \in \mathcal{D}' \times \mathcal{X}^{\leq n}$ is a forgery for h_v.

b) Let $\mathcal{X} := \{0,1\}$ and let $h\colon \mathcal{X}^{n+t} \to \mathcal{X}^n$ be a compression function, where $n, t \in \mathbb{N}$ such that $n \leq t$. Let $\widehat{h}\colon \mathcal{X}^* \to \mathcal{X}^n$ be the iterated hash function given in (20.2), using the key $0^n \in \mathcal{X}^n$. We consider the keyed hash function $\eta\colon \mathcal{X}^t \times \mathcal{X}^* \to \mathcal{X}^n\colon [u, w] \mapsto \widehat{h}(uw)$ and the associated nested keyed hash function $\mathcal{X}^t \times \mathcal{X}^* \to \mathcal{X}^n\colon [u, w] \mapsto \eta_u(\eta_u(w)) = \widehat{h}(u\widehat{h}(uw))$, taking the key $u \in \mathcal{X}^t$ twice.

Hence by the above we have to ensure that it is difficult to find collisions of the inner function $\eta_u\colon \mathcal{X}^* \to \mathcal{X}^n\colon w \mapsto \widehat{h}(uw) =: w'$, for all keys $u \in \mathcal{X}^t$; e. g. the Merkle-Damgard construction hints to how this might be achieved using variations of $\widehat{h}$ and suitable compression functions h to start with.

To compute the outer function $(\eta_u)|_{\mathcal{X}^n}\colon \mathcal{X}^n \to \mathcal{X}^n\colon w' \mapsto \widehat{h}(uw')$, since $l(uw') = t + n \leq 2t$ the argument is first padded giving $uw'\widehat{w} \in \mathcal{X}^{2t}$, where $l(\widehat{w}) = t - n$, to which then the compression function h is applied twice. Since $l(u) = t$ this yields $\widehat{h}(uw') = h(h(0^nu)w'\widehat{w})$, hence the first application of h only depends on u, and we have only one application of h depending on w'. Thus by the above we also have to ensure that it is difficult to find forgeries of $(\eta_u)|_{\mathcal{X}^n}$, which in turn means that the compression function h must be chosen suitably.

In practice, a proposed standard is the nested function HMAC [1996], which is based on the SHA-1 standard: We have SHA-1: $\mathcal{X}^{\leq m} \to \mathcal{X}^n$, where $\mathcal{X} := \{0,1\}$ and $m = 2^{64} - 1$ and $n = 160$, being based on a compression function $\mathcal{X}^{n+t} \to \mathcal{X}^n$ where $t = 512$. Letting $p, q, u \in \mathcal{X}^t$ we have $\mathrm{HMAC}_{p,q,u}(w) := \text{SHA-1}((u \oplus q) \cdot \text{SHA-1}((u \oplus p)w))$ for $w \in \mathcal{X}^{\leq m-t}$, where p, q are public and u is private. For further comments on MACs see [12, Ch.4.4] and [10, Ch.9.5].

(20.5) Strongly universal functions [Carter-Wegmann, 1979]. Let $\mathcal{X}$ be an alphabet, let $\mathcal{D}, \mathcal{H} \subseteq \mathcal{X}^*$ be finite, and let $h \colon \mathcal{H} \times \mathcal{D} \to \mathcal{X}^{\leq n}$, where $n \in \mathbb{N}$, be a keyed compression function. Let $\mathcal{H}$ be uniformly distributed.

a) Given $w \in \mathcal{D}$ and $v \in \mathcal{X}^{\leq n}$, the probability that $[w, v]$ is valid with respect to some key is given as $\mu(w, v) = \frac{|\{u \in \mathcal{H}; h_u(w) = v\}|}{|\mathcal{H}|}$. Hence a Las-Vegas algorithm finding a forgery without querying h, an **impersonation attack**, has a success probability which is bounded above by the **deception probability** $\epsilon_0 := \max\{\mu(w, v); [w, v] \in \mathcal{V}\}$, where $\mathcal{V} := \{[w, v] \in \mathcal{D} \times \mathcal{X}^{\leq n}; h_u(w) = v \text{ for some } u \in \mathcal{H}\}$.

For $w \in \mathcal{D}$ we have $\sum_{v \in \mathrm{im}(h)} \mu(w, v) = \sum_{v \in \mathrm{im}(h)} \frac{|\{u \in \mathcal{H}; h_u(w) = v\}|}{|\mathcal{H}|} = \frac{|\mathcal{H}|}{|\mathcal{H}|} = 1$, hence there is $v \in \mathrm{im}(h)$ such that $\mu(w, v) \geq \frac{1}{t}$, implying $\epsilon_0 \geq \frac{1}{t}$, and we have $\epsilon_0 = \frac{1}{t}$ if and only if for all $w \in \mathcal{D}$ and $v \in \mathrm{im}(h)$ we have $\mu(w, v) = \frac{1}{t}$.

b) Let $|\mathcal{D}| \geq 2$. Given $w \neq w' \in \mathcal{D}$ and $v, v' \in \mathcal{X}^{\leq n}$ such that $[w, v] \in \mathcal{V}$, the conditional probability that $[w', v']$ is valid with respect to some key $u \in \mathcal{H}$, such that $[w, v]$ also is valid with respect to the same key u, is given as $\mu(w', v' | w, v) = \frac{|\{u \in \mathcal{H}; h_u(w') = v', h_u(w) = v\}|}{|\{u \in \mathcal{H}; h_u(w) = v\}|}$. Hence a Las-Vegas algorithm finding a forgery querying h at most once, a **substitution attack**, has a success probability which is bounded above by the deception probability $\epsilon_1 := \max\{\mu(w', v' | w, v); [w', v'], [w, v] \in \mathcal{V}, w' \neq w\}$.

The keyed compression function h is called **strongly universal**, if for any $w \neq w' \in \mathcal{D}$ and $v, v' \in \mathrm{im}(h)$ we have $\frac{|\{u \in \mathcal{H}; h_u(w) = v, h_u(w') = v'\}|}{|\mathcal{H}|} = \frac{1}{t^2}$, where $t := |\mathrm{im}(h)| \in \mathbb{N}$. We have $\epsilon_1 \geq \frac{1}{t}$, with equality if and only if h is strongly universal; in the latter case we have $\epsilon_0 = \frac{1}{t}$ as well:

Again, for $w \neq w' \in \mathcal{D}$ and $v \in \mathrm{im}(h)$ such that $[w, v] \in \mathcal{V}$ we obtain $\sum_{v' \in \mathrm{im}(h)} \mu(w', v' | w, v) = \sum_{v' \in \mathrm{im}(h)} \frac{|\{u \in \mathcal{H}; h_u(w) = v, h_u(w') = v'\}|}{|\{u \in \mathcal{H}; h_u(w) = v\}|} = 1$, hence there is $v' \in \mathrm{im}(h)$ such that $\mu(w', v' | w, v) \geq \frac{1}{t}$ and thus $\epsilon_1 \geq \frac{1}{t}$. If $\epsilon_1 = \frac{1}{t}$, then for all $v' \in \mathrm{im}(h)$ we have $\mu(w', v' | w, v) \leq \frac{1}{t}$ and thus $\mu(w', v' | w, v) = \frac{1}{t}$. Hence we have $[w', \mathrm{im}(h)] \subseteq \mathcal{V}$, thus $\mathcal{V} = \mathcal{D} \times \mathrm{im}(h)$, implying $\mu(w', v' | w, v) = \mu(w, v | w', v')$, and thus $\mu(w, v) = \frac{1}{t}$, in particular $\epsilon_0 = \frac{1}{t}$. This yields $|\{u \in \mathcal{H}; h_u(w) = v, h_u(w') = v'\}| = |\mathcal{H}| \cdot \mu(w', v' | w, v) \mu(w, v) = \frac{|\mathcal{H}|}{t^2}$. If conversely h is strongly universal, then we have $|\{u \in \mathcal{H}; h_u(w) = v\}| = \sum_{v' \in \mathrm{im}(h)} |\{u \in \mathcal{H}; h_u(w) = v, h_u(w') = v'\}| = \sum_{v' \in \mathrm{im}(h)} \frac{|\mathcal{H}|}{t^2} = \frac{|\mathcal{H}|}{t}$, yielding $\mu(w', v' | w, v) = \frac{|\{u \in \mathcal{H}; h_u(w') = v', h_u(w) = v\}|}{|\{u \in \mathcal{H}; h_u(w) = v\}|} = \frac{|\mathcal{H}|}{t^2} \cdot \frac{t}{|\mathcal{H}|} = \frac{1}{t}$, thus $\epsilon_1 = \frac{1}{t}$. ♯

21 Signatures

(21.1) Signatures [Diffie-Hellman, 1976; Merkle, 1978]. a) Let $\mathcal{X}$ be an alphabet, let $\mathcal{D}, \mathcal{G} \subseteq \mathcal{X}^*$, and let $n \in \mathbb{N}$. A **signature scheme** or **digital signature** is a keyed hash function $g \colon \mathcal{G} \times \mathcal{D} \to \mathcal{X}^{\leq n} \colon [u, w] \mapsto g_u(w)$, called the **signature function**, together with a **(weak) verification function** $\gamma \colon \mathcal{G} \times \mathcal{X}^* \times \mathcal{X}^{\leq n} \to \{0, 1\} \colon [u, w, v] \mapsto \gamma_u(w, v)$ in FP, such that $\gamma_u(w, v) = 1$ whenever

$[w, v] \in \mathcal{X}^* \times \mathcal{X}^{\leq n}$ is valid for g with respect to the key $u \in \mathcal{G}$. If $\gamma_u(w, v) = 1$ implies that $[w, v]$ indeed is valid with respect to u then γ is called a **strong** verification function.

The key u used is kept private, while the verification function γ_u is made public. Hence there is a straightforward notion of **(existential or selective) forgeries** for signature schemes: These are pairs which either are valid but have been found without using the signature function, or are non-valid but verified by a weak verification function.

b) Signature schemes are often used in conjunction with hash functions: Let $\mathcal{X}$ be an alphabet, let $m, n \in \mathbb{N}$, and for $\mathcal{E} \subseteq \mathcal{X}^*$ let $h \colon \mathcal{E} \to \mathcal{X}^{\leq m}$ be a hash function. Let $g \colon \mathcal{G} \times \mathcal{D} \to \mathcal{X}^{\leq n}$ be a signature function such that $\operatorname{im}(h) \subseteq \mathcal{D} \subseteq \mathcal{X}^*$, with verification function $\gamma \colon \mathcal{G} \times \mathcal{X}^* \times \mathcal{X}^{\leq n} \to \{0, 1\}$. Then $hg \colon \mathcal{G} \times \mathcal{E} \to \mathcal{X}^{\leq n} \colon [u, w] \mapsto g_u(h(w))$ is a signature function, with verification function $\gamma(h) \colon \mathcal{G} \times \mathcal{X}^* \times \mathcal{X}^{\leq n} \to \{0, 1\} \colon [u, w, v] \mapsto \gamma_u(h(w), v)$.

We describe various attacks: For $w \in \mathcal{E}$ the selective forgery $[h(w), g_u(h(w))] \in \mathcal{X}^{\leq m} \times \mathcal{X}^{\leq n}$ for g_u yields the selective forgery $[w, g_u(h(w))] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ for hg. Hence it must be difficult to find selective forgeries for g_u.

Let $[w, v] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ be valid, hence $v = g_u(h(w))$, let $z := h(w) \in \mathcal{X}^{\leq m}$, and let $w \neq w' \in \mathcal{E}$ be a second preimage of z. Hence $[w', v] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ is an existential forgery found by a known-message attack. Hence it must be difficult to find second preimages under h.

Let $w \neq w' \in \mathcal{E}$ be a collision for h. If $h(w) \in \mathcal{X}^{\leq m}$ is signed using g_u, yielding $v := g_u(h(w)) \in \mathcal{X}^{\leq n}$, the pair $[w, v] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ is valid, and thus $[w', v] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ is an existential forgery found by a chosen-message attack. Hence it must be difficult to find collisions of h.

Let the signature scheme g be subject to existential forgery by a **key-only attack**, i. e. an attack only using the verification function γ, let $[z, v] \in \operatorname{im}(h) \times \mathcal{X}^{\leq n}$ be a forgery for g_u thus found, and let $w \in \mathcal{E}$ be a preimage of z. Thus $[w, v] \in \mathcal{E} \times \mathcal{X}^{\leq n}$ is an existential forgery. Hence if g is susceptible to key-only attacks, it must be difficult to find preimages under h.

(21.2) The Rivest-Shamir-Adleman (RSA) signature scheme [1978]. **a)** Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, let $\mathcal{D} := \mathbb{Z}/n\mathbb{Z}$ and $\mathcal{G} := (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$. For $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ let $d := e^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, where $[p, q, d]$ is the private key and $[n, e]$ is the public key, we get the signature function $g_e \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \colon x \mapsto x^d$, and the strong verification function $\gamma_e \colon (\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z}) \to \{0, 1\}$, where $\gamma_e(x, y) := 1$ if and only if $y^e = x \in \mathbb{Z}/n\mathbb{Z}$.

The security of the RSA signature scheme depends on the computational difficulty of integer factorisation. We describe various attacks, which can be prevented by using the signature scheme in conjunction with a hash function, or by requiring that messages are only chosen from a suitable admissible subset:

Letting $y \in \mathbb{Z}/n\mathbb{Z}$ and $x := y^e \in \mathbb{Z}/n\mathbb{Z}$ yields the existential forgery $[x, y]$

by a key-only attack. Letting $[x_1, y_1]$ and $[x_2, y_2]$ be valid, multiplicativity of the signature function yields the existential forgery $[x_1x_2, y_1y_2]$ by a known-message attack. Letting $x \in \mathbb{Z}/n\mathbb{Z}$ and $x_1 \in (\mathbb{Z}/n\mathbb{Z})^*$ and $x_2 := \frac{x}{x_1} \in \mathbb{Z}/n\mathbb{Z}$, we have $x = x_1x_2$ and querying for signatures $y_1 := g_e(x_1)$ and $y_2 := g_e(x_2)$ multiplicativity of the signature function yields the selective forgery $[x, y_1y_2]$ by a chosen-message attack.

b) Encryption and signatures can be combined as follows: A message is to be sent encrypted from Bob to Alice, and Alice has to be sure that Bob indeed is the sender and that the message has not been changed in between. To achieve this, let both Alice and Bob have RSA cryptosystems available, based on odd primes $p \neq q \in \mathbb{N}$ and $p' \neq q' \in \mathbb{N}$, respectively, where $n := pq \geq n' := p'q'$; hence we have $\mathbb{Z}_{n'} \subseteq \mathbb{Z}_n$. Let $e = d^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ and $e' = d'^{-1} \in (\mathbb{Z}/\varphi(n')\mathbb{Z})^*$ be Alice's and Bob's pairs of keys, respectively.

Let $x \in \mathbb{Z}_{n'} \subseteq \mathbb{Z}_n$. Bob first uses his signature function to compute $y := g_{e'}(x) \in \mathbb{Z}_{n'} \subseteq \mathbb{Z}_n$, and then uses Alice's encryption function to send the message $[E_e(x), E_e(y)] \in \mathbb{Z}_n^2$. Alice first decrypts it using her decryption function, yielding $[x, y] = [D_d(E_e(x)), D_d(E_e(y))] \in \mathbb{Z}_n^2$, and since $x, y \in \mathbb{Z}_{n'} \subseteq \mathbb{Z}_n$ she then uses Bob's verification function to check $\gamma_{e'}(x, y) = 1$.

c) Alternatively, we could also proceed the other way around: Letting $n \leq n'$, Bob first encrypts $x \in \mathbb{Z}_n \subseteq \mathbb{Z}_{n'}$ using Alice's encryption function, yielding $y := E_e(x) \in \mathbb{Z}_n \subseteq \mathbb{Z}_{n'}$, then uses his signature function to compute $z := g_{e'}(y) \in \mathbb{Z}_{n'}$, and sends the message $[y, z] \in \mathbb{Z}_{n'}^2$. Alice first uses Bob's verification function to check $\gamma_{e'}(y, z) = 1$, and then decrypts $y \in \mathbb{Z}_n \subseteq \mathbb{Z}_{n'}$ using her decryption function yielding $x = D_d(y)$.

This allows for the following **man-in-the-middle attack**, which shows that signing before encrypting is preferable: Let Oscar also have an RSA cryptosystem available, based on odd primes $p'' \neq q'' \in \mathbb{N}$ such that $n \leq n' \leq n'' := p''q''$, and let $e'' = d''^{-1} \in (\mathbb{Z}/\varphi(n')\mathbb{Z})^*$ be Oscar's pair of keys. Oscar intercepts a message $[y, z] \in \mathbb{Z}_{n'}^2$ sent from Bob, computes $\widetilde{z} := g_{e''}(y) \in \mathbb{Z}_{n''}$ using his signature function, and sends $[y, \widetilde{z}] \in \mathbb{Z}_{n''}^2$ instead to Alice. Then Alice uses Oscar's verification function to check $\gamma_{e''}(y, \widetilde{z}) = 1$, and thus concludes that the message originated from Oscar instead from Bob.

(21.3) Efficient Digital Signature (ESIGN) [Okamoto-Shiraishi, 1985]. Let $p \geq q \in \mathbb{N}$ be primes, let $n := p^2q$, let $r \in (\mathbb{Z}/n\mathbb{Z})^*$, and let $\mathcal{G} := \{e \in \mathbb{N}; \gcd(e, p) = 1\}$, where $[p, q, r]$ is the private key and $[n, e]$ is the public key. Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} := \mathcal{X}^*$, and let $h \colon \mathcal{X}^* \to \mathbb{Z}_n$ be a hash function. For $w \in \mathcal{X}^*$ let $x \in \mathbb{Z}_n$ such that $x = h(w) - r^e \in \mathbb{Z}/n\mathbb{Z}$, let $y := \lceil \frac{x}{pq} \rceil \cdot \frac{1}{er^{e-1}} \in \mathbb{Z}/p\mathbb{Z}$, and let $v := r + ypq \in \mathbb{Z}/n\mathbb{Z}$. Let the signature functions be defined as $g_e \colon \mathcal{X}^* \to \mathbb{Z}/n\mathbb{Z} \colon w \mapsto v$. The verification function $\gamma_e \colon \mathcal{X}^* \times (\mathbb{Z}/n\mathbb{Z}) \to \{0, 1\}$ is defined as follows: Given $[w, v] \in \mathcal{X}^* \times (\mathbb{Z}/n\mathbb{Z})$, let $z \in \mathbb{Z}_n$ such that $z = v^e \in \mathbb{Z}/n\mathbb{Z}$, and let $\gamma_e(w, v) := 1$ if and only if $h(w) \leq z \leq h(w) + 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$. Indeed valid pairs are verified: We have $z = v^e = (r + ypq)^e = \sum_{i=0}^{e} \binom{e}{i} r^{e-i}(ypq)^i = r^e + er^{e-1}ypq \in \mathbb{Z}/n\mathbb{Z}$. Since $er^{e-1}y = \lceil \frac{x}{pq} \rceil \in \mathbb{Z}/p\mathbb{Z}$ this yields $z = h(w) - x + pq\lceil \frac{x}{pq} \rceil \in \mathbb{Z}/n\mathbb{Z}$, and thus $h(w) \leq z \leq h(w) + pq$, where $pq \leq n^{\frac{2}{3}} \leq 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$.

Forgeries can be found by a known-message attack as follows: Let $[w, v] \in \mathcal{X}^* \times (\mathbb{Z}/n\mathbb{Z})$ be valid. If $w' \in \mathcal{X}^*$ such that $h(w') \leq z \leq h(w') + 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$, where $z \in \mathbb{Z}_n$ such that $z = v^e \in \mathbb{Z}/n\mathbb{Z}$, then $[w', v]$ is a forgery. Hence it must be difficult to find preimages under h of w' such that $|h(w') - h(w)| \leq 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$.

Forgeries can also be found by a chosen-message attack as follows: Let $w \in \mathcal{X}^*$, let $v := g_e(w) \in \mathbb{Z}/n\mathbb{Z}$, and let $z \in \mathbb{Z}_n$ such that $z = v^e \in \mathbb{Z}/n\mathbb{Z}$, hence $h(w) \leq z \leq h(w) + 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$. If $w' \in \mathcal{X}^*$ such that $h(w) \leq h(w') \leq z$, we have $h(w') \leq z \leq h(w) + 2^{\lceil \frac{2}{3} \log_2(n) \rceil} \leq h(w') + 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$, thus $[w', v]$ is a forgery. From $h(w) \leq h(w') \leq z \leq h(w) + 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$ we get $|h(w') - h(w)| \leq 2^{\lceil \frac{2}{3} \log_2(n) \rceil}$, i. e. $h(w)$ and $h(w')$ coincide in the $\lceil \frac{1}{3} \log_2(n) \rceil$ highest of their $b_2(n-1) = \lceil \log_2(n) \rceil$ bits. Hence by way of a birthday attack such $w, w' \in \mathcal{X}^*$ are expected to be found using $O(n^{\frac{1}{6}})$ queries of h.

It has been shown that the ESIGN signature scheme can be broken if $e \leq 3$, hence we have to choose $e \geq 4$. The security of the ESIGN signature scheme also depends on the computational difficulty of integer factorisation, where it is an open problem whether factoring integers of the form $p^2 q$ is as difficult than factoring integers of the form pq. For practical purposes, the ESIGN signature scheme runs much faster than the RSA signature scheme.

(21.4) Remark. Other signature schemes using RSA moduli, and thus depending on the computational difficulty of integer factorisation are the **Rabin signature scheme**, based on the Rabin cryptosystem, see [10, Ch.11.3.4], the **Feige-Fiat-Shamir signature scheme**, based on the Fiat-Shamir identification scheme (23.3), see [10, Ch.11.4.1], and the **Guillou-Quisquater signature scheme**, based on the Guillou-Quisquater identification scheme (23.4), see [10, Ch.11.4.2]; the general idea of converting a challenge-response identification scheme into a signature scheme is to replace the challenge by a hash value of the concatenation of the witness and the message to be signed, and the corresponding response is the signature.

(21.5) The ElGamal signature scheme [1985]. a) Let $p \in \mathbb{N}$ be a prime, and let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root. For $[a, b] \in \mathcal{G} := (\mathbb{Z}/(p-1)\mathbb{Z}) \times (\mathbb{Z}/(p-1)\mathbb{Z})^*$ let $\alpha := \rho^a \in (\mathbb{Z}/p\mathbb{Z})^*$, and $\beta \in \mathbb{Z}_p^*$ such that $\beta = \rho^b \in (\mathbb{Z}/p\mathbb{Z})^*$; hence $\beta \in (\mathbb{Z}/p\mathbb{Z})^*$ is a primitive root. Let $[p, \rho, \alpha]$ be the public key, and $[a, b]$ be the private key. Letting $\mathcal{D} := \mathbb{Z}/(p-1)\mathbb{Z}$ we get the signature function $g_{a,b} \colon \mathbb{Z}/(p-1)\mathbb{Z} \to \mathbb{Z}_p^* \times (\mathbb{Z}/(p-1)\mathbb{Z}) \colon x \mapsto [\beta, \frac{x - a\beta}{b}]$.

The verification function $\gamma_{a,b} \colon (\mathbb{Z}/(p-1)\mathbb{Z}) \times \mathbb{Z}_p^* \times (\mathbb{Z}/(p-1)\mathbb{Z}) \to \{0, 1\}$ is given by $\gamma_{a,b}(x, \zeta, y) := 1$ if and only if $\alpha^\zeta \zeta^y = \rho^x \in (\mathbb{Z}/p\mathbb{Z})^*$: Letting $z := \log_\rho(\zeta) \in \mathbb{Z}/(p-1)\mathbb{Z}$, we have $\alpha^\zeta \zeta^y = (\rho^a)^\zeta (\rho^z)^y = \rho^{a\zeta + zy} = \rho^x \in (\mathbb{Z}/p\mathbb{Z})^*$ if and only if $a\zeta + zy = x \in \mathbb{Z}/(p-1)\mathbb{Z}$, which holds if and only if $y = \frac{x - a\zeta}{z} \in \mathbb{Z}/(p-1)\mathbb{Z}$. The latter equation indeed holds true for a valid pair $[x; \beta, \frac{x - a\beta}{b}]$.

b) The security of the ElGamal signature scheme depends on the computational difficulty of the discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$. Indeed, given $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ selective forgeries are difficult to find by way of key-only attacks: If a primitive root $\beta \in \mathbb{Z}_p^*$ is chosen first, then $y = \log_\beta(\rho^x \alpha^{-\beta}) \in \mathbb{Z}/(p-1)\mathbb{Z}$ is the solution of a discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$. If $y \in \mathbb{Z}/(p-1)\mathbb{Z}$ is chosen first, the equation $\alpha^\beta \beta^y = \rho^x \in (\mathbb{Z}/p\mathbb{Z})^*$ has to be solved for $\beta \in \mathbb{Z}_p^*$, an efficient solution not being known. Finally $[\beta, y]$ can be computed simultaneously, where again no efficient solution is known. Actually, selective forgeries seem to be difficult to find by any type of attack, hence the ElGamal signature scheme is considered to be secure in conjunction with a sufficiently secure hash function.

If $[\beta, y]$ is given, then $x = \log_\rho(\alpha^\beta \beta^y) \in \mathbb{Z}/(p-1)\mathbb{Z}$ again is the solution of a discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$, thus existential forgeries with prescribed signature are difficult to find by way of key-only attacks. But usual existential forgeries using a key-only attack are possible: Let $i \in \mathbb{Z}/(p-1)\mathbb{Z}$ and $j \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, and let $\beta \in \mathbb{Z}_p^*$ such that $\beta = \rho^i \alpha^j \in (\mathbb{Z}/p\mathbb{Z})^*$. Hence $\alpha^\beta \beta^y = \rho^x \in (\mathbb{Z}/p\mathbb{Z})^*$ is equivalent to $\alpha^{\beta+jy} = \rho^{x-iy} \in (\mathbb{Z}/p\mathbb{Z})^*$, which holds if both $\beta + jy = 0 \in \mathbb{Z}/(p-1)\mathbb{Z}$ and $x - iy = 0 \in \mathbb{Z}/(p-1)\mathbb{Z}$, where the latter system of equations is equivalent to $y = \frac{-\beta}{j} \in \mathbb{Z}/(p-1)\mathbb{Z}$ and $x = \frac{-i\beta}{j} \in \mathbb{Z}/(p-1)\mathbb{Z}$. Hence we have found the forgery $[\frac{-i\beta}{j}; \beta, \frac{-\beta}{j}]$.

Existential forgeries using a known-message attack are possible as well: Let $[x; \beta, y]$ be valid, let $i, j, k \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $k\beta - jy \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, let $\beta' \in \mathbb{Z}_p^*$ such that $\beta' = \rho^i \alpha^j \beta^k \in (\mathbb{Z}/p\mathbb{Z})^*$, and let $y' := \frac{y\beta'}{k\beta - jy} \in \mathbb{Z}/(p-1)\mathbb{Z}$ and $x' := \frac{\beta'(kx+iy)}{k\beta - jy} \in \mathbb{Z}/(p-1)\mathbb{Z}$. Then $[x'; \beta', y']$ is a forgery: The condition $\alpha^{\beta'} \beta'^{y'} = \rho^{x'} \in (\mathbb{Z}/p\mathbb{Z})^*$ is equivalent to $\alpha^{\beta'+jy'} \beta^{ky'} = \rho^{x'-iy'} \in (\mathbb{Z}/p\mathbb{Z})^*$. Since $\beta' + jy' = \frac{k\beta\beta'}{k\beta - jy} \in \mathbb{Z}/(p-1)\mathbb{Z}$, the left hand side of the latter condition is $\alpha^{\beta'+jy'} \beta^{ky'} = \alpha^{\frac{k\beta\beta'}{k\beta - jy}} \beta^{\frac{ky\beta'}{k\beta - jy}} \in (\mathbb{Z}/p\mathbb{Z})^*$, while since $x' - iy' = \frac{\beta' kx}{k\beta - jy} \in \mathbb{Z}/(p-1)\mathbb{Z}$ and $\alpha^\beta \beta^y = \rho^x \in (\mathbb{Z}/p\mathbb{Z})^*$ the right hand side equals $\rho^{x'-iy'} = (\rho^x)^{\frac{\beta' k}{k\beta - jy}} = \alpha^{\frac{k\beta\beta'}{k\beta - jy}} \beta^{\frac{ky\beta'}{k\beta - jy}} \in (\mathbb{Z}/p\mathbb{Z})^*$.

c) We consider security objectives related to the private key $[a, b]$: Breaking the ElGamal signature scheme amounts to finding the key a, where finding $a := \log_\rho(\alpha)$ from α alone amounts to solve a discrete logarithm problem in $(\mathbb{Z}/p\mathbb{Z})^*$. Let $[x; \beta, y] \in \mathbb{Z}_{p-1} \times \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ be valid. If $y = 0$ then $a \in \mathbb{Z}_{p-1}$ can be determined from $a\beta = x - by = x \in \mathbb{Z}/(p-1)\mathbb{Z}$, hence we may assume that $y \neq 0$; if $a \in \mathbb{Z}_{p-1}$ is known, then $b \in \mathbb{Z}_{p-1}$ can be determined from $a\beta = x - by \in \mathbb{Z}/(p-1)\mathbb{Z}$ again. Conversely, if $b \in \mathbb{Z}_{p-1}$ is known, then $a \in \mathbb{Z}_{p-1}$ can be computed as follows: For $d := \gcd(p-1, \beta)$ let $p' := \frac{p-1}{d}$ and $\beta' := \frac{\beta}{d}$, and since $a\beta = x - by \in \mathbb{Z}/(p-1)\mathbb{Z}$ let $x' := \frac{x-by}{d} \in \mathbb{Z}$. This yields $a = \frac{x'}{\beta'} \in \mathbb{Z}/p'\mathbb{Z}$, and letting $z' \in \mathbb{Z}_{p'}$ such that $z' = \frac{x'}{\beta'} \in \mathbb{Z}/p'\mathbb{Z}$ we obtain $a = z' + ip' \in \mathbb{Z}/(p-1)\mathbb{Z}$ for some $i \in \{0, \ldots, d-1\}$. Thus a can be found by checking the condition $\rho^{\frac{x'}{\beta'} + ip'} = \rho^a = \alpha \in (\mathbb{Z}/p\mathbb{Z})^*$ for $i \in \{0, \ldots, d-1\}$, which is feasible if d is small.

If a is kept fixed, b must only be used once; hence by varying b the ElGamal signature scheme is used as a randomised signature scheme: Let $[x_1; \beta, y_1] \in \mathbb{Z}_{p-1} \times \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ and $[x_2; \beta, y_2] \in \mathbb{Z}_{p-1} \times \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ be valid with respect to $[a, b]$. Since $\alpha^\beta \beta^{y_i} = \rho^{x_i} \in (\mathbb{Z}/p\mathbb{Z})^*$ we have $\rho^{x_1 - x_2} = \beta^{y_1 - y_2} = \rho^{b(y_1 - y_2)} \in (\mathbb{Z}/p\mathbb{Z})^*$, which is equivalent to $(x_1 - x_2) = b(y_1 - y_2) \in \mathbb{Z}/(p-1)\mathbb{Z}$. For $d := \gcd(p-1, y_1 - y_2)$ let $p' := \frac{p-1}{d}$, and let $y' := \frac{y_1 - y_2}{d}$ and $x' := \frac{x_1 - x_2}{d} \in \mathbb{Z}$. Since $x' = by' \in \mathbb{Z}/p'\mathbb{Z}$ we let $z' \in \mathbb{Z}_{p'}$ such that $z' = \frac{x'}{y'} \in \mathbb{Z}/p'\mathbb{Z}$, hence we obtain $b = z' + ip' \in \mathbb{Z}/(p-1)\mathbb{Z}$ for some $i \in \{0, \ldots, d-1\}$. Hence b is found by checking the condition $\rho^{z' + ip'} = \rho^b = \beta \in (\mathbb{Z}/p\mathbb{Z})^*$ for $i \in \{0, \ldots, d-1\}$.

(21.6) The Schnorr signature scheme [1991]. Given a prime modulus p the ElGamal signature scheme yields an output of bit length $2b_2(p)$, where the bit length of the public key is not counted. Since for security reasons $b_2(p)$ cannot be chosen too small, for practical applications a shorter signature is desirable. To this end, the Schnorr signature scheme varies the ElGamal signature scheme, making use of a second smaller prime q, and integrating a hash function.

Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, and let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity; σ can be found from a primitive root $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ as $\sigma := \rho^{\frac{p-1}{q}} \in (\mathbb{Z}/p\mathbb{Z})^*$. For $[a, b] \in \mathcal{G} := (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ let $\alpha := \sigma^a \in (\mathbb{Z}/p\mathbb{Z})^*$, and $\beta \in \mathbb{Z}_p^*$ such that $\beta = \sigma^b \in (\mathbb{Z}/p\mathbb{Z})^*$. Let $[p, q, \sigma, \alpha]$ be the public key, and $[a, b]$ be the private key. Integers are considered to be given in their binary representation; for the transition from integers to binary representations a **redundancy function** might be used, where security objectives related to the latter are not discussed here. Let $\mathcal{X} := \{0, 1\}$ and let $h\colon \mathcal{X}^* \to \mathbb{Z}/q\mathbb{Z}$ be a hash function. For $x \in \mathcal{X}^*$ let $y := h(x\beta) \in \mathbb{Z}/q\mathbb{Z}$. Letting $\mathcal{D} := \mathcal{X}^*$ we get the signature function $g_{a,b}\colon \mathcal{X}^* \to (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})\colon : x \mapsto [y, b + ay]$.

The verification function $\gamma_{a,b}\colon \mathcal{X}^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \to \{0, 1\}$ is given as follows: For $[x, y, z] \in \mathcal{X}^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ let $v \in \mathbb{Z}_p$ such that $v = \sigma^z \alpha^{-y} \in (\mathbb{Z}/p\mathbb{Z})^*$, and let $\gamma_{a,b}(x, y, z) := 1$ if and only if $h(xv) = y$. Indeed, if $[x; y, z]$ is valid then we have $\sigma^z \alpha^{-y} = \sigma^{b+ay} \sigma^{-ay} = \sigma^b = \beta \in (\mathbb{Z}/p\mathbb{Z})^*$, hence $v = \beta$ and $h(xv) = h(x\beta) = y$. Because of the use of h we cannot expect to have strong verification functions, for practical purposes we have to assume that for given $[y, z]$ it is difficult to find preimages under h of y having $v = \sigma^z \alpha^{-y}$ as a postfix.

(21.7) The Nyberg-Rueppel signature scheme [1993]. The schemes given so far are signature schemes **with appendix**, i. e. messages are required as arguments of the verification functions. The following variant of the Schnorr signature scheme allows for **message recovery**, i. e. messages can be recovered from their signatures; hence to prevent existential forgeries only messages from an admissible subset may be allowed:

Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, and let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity. For $[a, b] \in \mathcal{G} := (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ let $\alpha := \sigma^a \in (\mathbb{Z}/p\mathbb{Z})^*$ and $\beta := \sigma^b \in (\mathbb{Z}/p\mathbb{Z})^*$. Let $[p, q, \sigma, \alpha]$ be the public key, and $[a, b]$ be the private key. For $x \in (\mathbb{Z}/p\mathbb{Z})^*$ let $y \in \mathbb{Z}_p^*$ such that $y = \frac{x}{\beta} \in (\mathbb{Z}/p\mathbb{Z})^*$. Letting $\mathcal{D} := (\mathbb{Z}/p\mathbb{Z})^*$ we get the signature function $g_{a,b} \colon (\mathbb{Z}/p\mathbb{Z})^* \to \mathbb{Z}_p^* \times (\mathbb{Z}/q\mathbb{Z}) \colon x \mapsto [y, b + ay]$. The verification function $\gamma_{a,b} \colon (\mathbb{Z}/p\mathbb{Z})^* \times \mathbb{Z}_p^* \times (\mathbb{Z}/q\mathbb{Z}) \to \{0,1\}$ is given as follows: For $[x,y,z] \in (\mathbb{Z}/p\mathbb{Z})^* \times \mathbb{Z}_p^* \times (\mathbb{Z}/q\mathbb{Z})$ let $\gamma_{a,b}(x,y,z) := 1$ if and only if $x = y\sigma^z\alpha^{-y} \in (\mathbb{Z}/p\mathbb{Z})^*$. Indeed, if $[x;y,z]$ is valid then we have $y\sigma^z\alpha^{-y} = y\sigma^{b+ay}\sigma^{-ay} = y\sigma^b = \frac{x}{\beta} \cdot \beta = x \in (\mathbb{Z}/p\mathbb{Z})^*$.

(21.8) The Digital Signature Algorithm (DSA) [1994]. a) The DSA is a currently used standard for practical applications. It is a combination of the ElGamal and the Schnorr signature schemes, where $b_2(q) = 160$ and $64 \mid b_2(p)$ such that $512 \leq b_2(p) \leq 1024$, the maximum being recommended, using SHA-1: $\mathcal{X}^{\leq m} \to \mathcal{X}^n$ as hash function, where $\mathcal{X} := \{0,1\}$ and $m = 2^{64} - 1$ and $n = 160$.

Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, and let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity. For $[a,b] \in \mathcal{G} := (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})^*$ let $\alpha := \sigma^a \in (\mathbb{Z}/p\mathbb{Z})^*$, and $\beta \in \mathbb{Z}_p^*$ such that $\beta = \sigma^b \in (\mathbb{Z}/p\mathbb{Z})^*$. Let $[p,q,\sigma,\alpha]$ be the public key, and $[a,b]$ be the private key. Let $\mathcal{X} := \{0,1\}$ and let $h \colon \mathcal{X}^* \to \mathbb{Z}/q\mathbb{Z}$ be a hash function. Letting $\mathcal{D} \subseteq \mathcal{X}^*$ be chosen such that $\frac{h(x)+a\beta}{b} \neq 0 \in \mathbb{Z}/q\mathbb{Z}$ we get the signature function $g_{a,b} \colon \mathcal{D} \to (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})^* \colon x \mapsto [\beta, \frac{h(x)+a\beta}{b}]$.

The verification function $\gamma_{a,b} \colon \mathcal{X}^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})^* \to \{0,1\}$ is defined as follows: For $[x,y,z] \in \mathcal{X}^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})^*$ let $w \in \mathbb{Z}_p^*$ such that $w = \sigma^{\frac{h(x)}{z}}\alpha^{\frac{y}{z}} \in (\mathbb{Z}/p\mathbb{Z})^*$, where the exponents are considered as elements of $\mathbb{Z}/q\mathbb{Z}$, and let $\gamma_{a,b}(x,y,z) := 1$ if and only if $w = y \in \mathbb{Z}/q\mathbb{Z}$. If $[x;y,z]$ is valid then we have $\sigma^{\frac{h(x)}{z}}\alpha^{\frac{y}{z}} = \sigma^{\frac{b(h(x)+a\beta)}{h(x)+a\beta}} = \sigma^b = \beta \in (\mathbb{Z}/p\mathbb{Z})^*$, hence we have $w = \beta$.

b) Another standard currently used is the **Elliptic Curve Digital Signature Algorithm (ECDSA)** [2000], which is based on the computational difficulty of the discrete logarithm problem for elliptic curves, and also is a combination of the ElGamal and the Schnorr signature schemes; see [12, Ch.7.4.3].

22 Undeniable signatures

(22.1) The Lamport signature scheme [1976]. a) Let $\mathcal{X} := \{0,1\}$ and let $f \colon \mathcal{C} \to \mathcal{X}^{\leq n}$ be a compression function for some $n \in \mathbb{N}$; in particular $\mathcal{C}$ is finite. Let $k \in \mathbb{N}$, and for $i \in \{1,\ldots,k\}$ and $j \in \{0,1\}$ choose $u_{i,j} \in \mathcal{C}$, let $v_{i,j} := f(u_{i,j}) \in \mathcal{X}^{\leq n}$, and let $u := [u_{1,0}, u_{1,1}, \ldots, u_{k,1}] \in \mathcal{C}^{2k}$ be the private key and $v := [v_{1,0}, v_{1,1}, \ldots, v_{k,1}] \in (\mathcal{X}^{\leq n})^{2k}$ be the public key. Let the signature functions be defined as $g_u \colon \mathcal{X}^k \to \mathcal{C}^k \colon x_1 \cdots x_k \mapsto [u_{1,x_1}, \ldots, u_{k,x_k}]$. For $[u_1, \ldots, u_k] \in \mathcal{C}^k$ let the verification function $\gamma_u \colon \mathcal{X}^k \times \mathcal{C}^k \to \{0,1\}$ be defined as $\gamma_u(x_1 \cdots x_k; u_1, \ldots, u_k) := 1$ if and only if $f(u_i) = v_{i,x_i}$ for $i \in \{1,\ldots,k\}$: Indeed for $[x_1 \cdots x_k; u_{1,x_1}, \ldots, u_{k,x_k}]$ we have $f(u_{i,x_i}) = v_{i,x_i}$ for $i \in \{1,\ldots,k\}$. Because of the use of the compression function f we cannot expect to have strong verification functions, for practical considerations we thus have to assume that it is difficult to find preimages under f.

If a key is used more than once, existential forgeries can be computed by a known-message attack; hence this is a **one-time signature scheme**: For $k \geq 2$ let $[x_1 \cdots x_k; u_{1,x_1}, \ldots, u_{k,x_k}]$ and $[x'_1 \cdots x'_k; u_{1,x'_1}, \ldots, u_{k,x'_k}]$ be valid such that $x_1 \cdots x_k$ and $x'_1 \cdots x'_k$ differ in positions $i < j$. Hence $u_{i,0}, u_{i,1}$ and $u_{j,0}, u_{j,1}$ are known, and $[x_1 \cdots x'_i \cdots x'_j \cdots x_k; u_{1,x_1}, \ldots, u_{i,x'_i}, \ldots, u_{j,x'_j}, \ldots, u_{k,x_k}]$ is valid.

b) We show that finding preimages under f reduces to finding existential forgeries by a key-only attack: Let $f\colon \mathcal{C} \rightarrow \mathcal{C}$ be bijective, and assume there is an oracle finding an existential forgery for each public key consisting of $2k \leq |\mathcal{C}|$ pairwise distinct elements. Then we have the following Las-Vegas algorithm to find the preimage under f of $v \in \mathcal{C}$: Choose $i \in \{1, \ldots, k\}$ and $j \in \{0, 1\}$, and choose a public key $[v_{1,0}, v_{1,1}, \ldots, v_{k,1}] \in \mathcal{C}^{2k}$ consisting of $2k$ pairwise distinct elements such that $v_{i,j} = v$, and let $[x_1 \cdots x_k; u_1, \ldots, u_k]$ be an existential forgery. If $x_i = j$ then we indeed have $f(u_i) = v_{i,x_i} = v_{i,j} = v$.

We show that this algorithm has success probability at least $\frac{1}{2}$: Let $\mathcal{V}$ be the set of all public keys consisting of $2k$ pairwise distinct elements, let $\mathcal{V}_v \subseteq \mathcal{V}$ be the subset of the keys containing v, and let $\mathcal{V}'_v \subseteq \mathcal{V}_v$ in turn be the subset of the keys such that the existential forgery found fulfils $x_i = j$ where $v_{i,j} = v$. We firstly consider the matrix in $\{0, 1\}^{\mathcal{V} \times \mathcal{C}}$, whose rows and columns are indexed by $\mathcal{V}$ and $\mathcal{C}$, respectively, and whose $[v_{1,0}, v_{1,1}, \ldots, v_{k,1}; v]$-entry is 1 if and only if $[v_{1,0}, v_{1,1}, \ldots, v_{k,1}]$ contains v. Hence each row contains precisely $2k$ non-vanishing entries, and all columns contain the same number of non-vanishing entries. Thus summing all matrix entries, adding rows first yields $2k|\mathcal{V}|$, and adding columns first yields $\sum_{w \in \mathcal{C}} |\mathcal{V}_w| = |\mathcal{C}| \cdot |\mathcal{V}_v|$, hence we have $2k|\mathcal{V}| = |\mathcal{C}| \cdot |\mathcal{V}_v|$.

We secondly consider the matrix in $\{0, 1\}^{\mathcal{V} \times \mathcal{C}}$, whose $[v_{1,0}, v_{1,1}, \ldots, v_{k,1}; v]$-entry is 1 if and only if $v_{i,j} = v$ and $x_i = j$ for some $i \in \{1, \ldots, k\}$ and $j \in \{0, 1\}$. Hence each row contains precisely k non-vanishing entries, thus summing all matrix entries, adding rows first yields $k|\mathcal{V}|$, and adding columns first yields $\sum_{w \in \mathcal{C}} |\mathcal{V}'_w|$. Thus the success probability of the above algorithm is given as $\frac{1}{|\mathcal{C}|} \cdot \sum_{v \in \mathcal{C}} \frac{|\mathcal{V}'_v|}{|\mathcal{V}_v|} = \frac{1}{2k|\mathcal{V}|} \cdot \sum_{v \in \mathcal{C}} |\mathcal{V}'_v| = \frac{k|\mathcal{V}|}{2k|\mathcal{V}|} = \frac{1}{2}$. ♯

(22.2) The Chaum-van Antwerpen signature scheme [1990]. Undeniable signatures require the signer to participate in the verification process, and prevent unacknowledged use of signed messages and the signer from disavowing.

a) Let $q < p \in \mathbb{N}$ be primes such that $q \mid p - 1$, let $e \in (\mathbb{Z}/q\mathbb{Z})^*$ and $d := e^{-1} \in (\mathbb{Z}/q\mathbb{Z})^*$, let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity, and let $\tau := \sigma^e \in (\mathbb{Z}/p\mathbb{Z})^*$. Let $[p, q, \sigma, \tau]$ be the public key, and $[e, d]$ be the private key. We have the bijective signature function $g_e\colon \langle\sigma\rangle \rightarrow \langle\sigma\rangle\colon x \mapsto x^e$. The verification function $\gamma_e\colon \langle\sigma\rangle \times \langle\sigma\rangle \rightarrow \{0, 1\}$ is defined by the following two-pass **challenge-response protocol**: Given $[x, y] \in \langle\sigma\rangle \times \langle\sigma\rangle$, Bob chooses $a \in (\mathbb{Z}/q\mathbb{Z})^*$ and $b \in \mathbb{Z}/q\mathbb{Z}$ randomly, and sends the **challenge** $z := y^a\tau^b \in \langle\sigma\rangle$ to Alice, who returns the **response** $w := z^d \in \langle\sigma\rangle$, and we let $\gamma_e(x, y) := 1$ if and only if $w = x^a\sigma^b \in \langle\sigma\rangle$. Indeed, if $[x, y]$ is valid, then we have $y = x^e$, thus $x = y^d$, and from $\tau = \sigma^e$ we get $\sigma = \tau^d$, implying $x^a\sigma^b = (y^a\tau^b)^d = z^d = w$.

This is only weak verification, but if $[x,y]$ is not valid, i. e. we have $y \neq x^e$, then w is verified, i. e. we have $w = x^a \sigma^b$, with a probability of at most $\frac{1}{q-1}$: Since $\langle \tau \rangle = \langle \sigma \rangle$, we for fixed $a \in (\mathbb{Z}/q\mathbb{Z})^*$ have $\{y^a \tau^b; b \in \mathbb{Z}/q\mathbb{Z}\} = \langle \sigma \rangle$. Thus each challenge z arises from precisely $q-1$ pairs $[a,b]$. Let $\alpha, \beta, \gamma, \delta \in \mathbb{Z}/q\mathbb{Z}$ such that $z = \sigma^\alpha$ and $w = \sigma^\beta$ and $x = \sigma^\gamma$ and $y = \sigma^\delta$. From $\sigma^\alpha = z = y^a \tau^b = \sigma^{\delta a + eb}$ and $\sigma^\beta = w = x^a \sigma^b = \sigma^{\gamma a + b}$ we obtain the system of linear equations $\alpha = \delta a + eb$ and $\beta = \gamma a + b$ for $[a,b] \in (\mathbb{Z}/q\mathbb{Z})^2$. From $\sigma^\delta = y \neq x^e = \sigma^{\gamma e}$ we conclude $\delta \neq \gamma e$. Hence the matrix of the above system has determinant $\delta - \gamma e \neq 0$, thus has a unique solution in $(\mathbb{Z}/q\mathbb{Z})^2$. Hence for a given challenge z there is at most one pair $[a,b]$, out of the $q-1$ pairs fulfilling $a \neq 0$ and yielding this challenge, such that the response w is verified.

b) The **disavowal protocol** is essentially a twofold run of the above challenge-response protocol: Bob chooses $a \in (\mathbb{Z}/q\mathbb{Z})^*$ and $b \in \mathbb{Z}/q\mathbb{Z}$ randomly, and sends the challenge $z := y^a \tau^b \in \langle \sigma \rangle$ to Alice, who returns the response $w := z^d \in \langle \sigma \rangle$. Then Bob checks whether $w = x^a \sigma^b \in \langle \sigma \rangle$, in which case Bob accepts $[x,y]$. Otherwise Bob chooses $a' \in (\mathbb{Z}/q\mathbb{Z})^*$ and $b' \in \mathbb{Z}/q\mathbb{Z}$ randomly, and sends the challenge $z' := y^{a'} \tau^{b'} \in \langle \sigma \rangle$ to Alice, who returns the response $w' := z'^d \in \langle \sigma \rangle$. Then Bob checks whether $w' = x^{a'} \sigma^{b'} \in \langle \sigma \rangle$, in which case Bob accepts $[x,y]$. Otherwise Bob accepts $[x,y]$ if and only if $(w\sigma^{-b})^{a'} \neq (w'\sigma^{-b'})^a \in \langle \sigma \rangle$.

If Alice honestly follows the disavowal protocol and $y \neq x^e$, then with a small probability Bob incorrectly accepts $[x,y]$ in the first or second step, or finally we have $(w\sigma^{-b})^{a'} = ((y^a \tau^b)^d \sigma^{-b})^{a'} = y^{aa'd} \tau^{dba'} \sigma^{-ba'} = y^{aa'd}$ and $(w'\sigma^{-b'})^a = ((y^{a'} \tau^{b'})^d \sigma^{-b'})^a = y^{aa'd} \tau^{db'a} \sigma^{-b'a} = y^{aa'd}$, thus Bob correctly rejects $[x,y]$.

If Alice is dishonest and does not follow the disavowal protocol, and $y = x^e$, then $w \neq x^a \sigma^b$ or $w' \neq x^{a'} \sigma^{b'}$. Then we have $(w\sigma^{-b})^{a'} = (w'\sigma^{-b'})^a$ with a probability of at most $\frac{1}{q-1}$, thus with a high probability Bob accepts $[x,y]$: Since $a \in (\mathbb{Z}/q\mathbb{Z})^*$ there is $\widetilde{a} \in (\mathbb{Z}/q\mathbb{Z})^*$ such that $a\widetilde{a} = 1$, hence $(w\sigma^{-b})^{a'} = (w'\sigma^{-b'})^a$ implies $w' = (w\sigma^{-b})^{\widetilde{a}a'} \sigma^{b'}$. Letting $x' := (w\sigma^{-b})^{\widetilde{a}} \in \langle \sigma \rangle$ we have $x'^{a'} \sigma^{b'} = w'$. If $w \neq x^a \sigma^b$ then $x \neq (w\sigma^{-b})^{\widetilde{a}} = x'$, while if $x'^{a'} \sigma^{b'} = w' \neq x^{a'} \sigma^{b'}$ then again $x' \neq x$. Hence we have $y = x^e \neq x'^e$, thus $[x', y]$ is not valid. Since $x'^{a'} \sigma^{b'} = w'$ the response w' is verified, which happens with a probability of at most $\frac{1}{q-1}$. $\sharp$

(22.3) The Van Heijst-Pedersen signature scheme [1992]. Fail-stop signatures provide a mechanism to prove that a signature is a forgery.

a) Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, let $e \in (\mathbb{Z}/q\mathbb{Z})^*$, let $\sigma \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity, let $\tau := \sigma^e \in (\mathbb{Z}/p\mathbb{Z})^*$, let $a, b, a', b' \in \mathbb{Z}/q\mathbb{Z}$ be chosen randomly, and let $t := \sigma^a \tau^b \in \langle \sigma \rangle$ and $t' := \sigma^{a'} \tau^{b'} \in \langle \sigma \rangle$. Let $[p, q, \sigma, \tau, t, t']$ be the public key, and $[a, b, a', b']$ be the private key, while the key e is chosen by a **trusted authority** and kept completely private, even from Alice and Bob; thus we have to assume that solving the discrete logarithm problem in $C_q \cong \langle \sigma \rangle \leq (\mathbb{Z}/p\mathbb{Z})^*$ is difficult.

Given $x \in \mathbb{Z}/q\mathbb{Z}$, let $y := a + xa' \in \mathbb{Z}/q\mathbb{Z}$ and $z := b + xb' \in \mathbb{Z}/q\mathbb{Z}$, yielding the signature function $g_{t,t'} \colon \mathbb{Z}/q\mathbb{Z} \to (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \colon x \mapsto [y, z]$. The verification function $\gamma_{t,t'} \colon (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \to \{0,1\}$ is defined by letting $\gamma_{t,t'}(x;y,z) := 1$ if and only if $tt'^x = \sigma^y\tau^z \in \langle\sigma\rangle$. Indeed, if $[x;y,z]$ is valid then we have $tt'^x = \sigma^{a+xa'}\tau^{b+xb'} = \sigma^y\tau^z \in \langle\sigma\rangle$. This is only weak verification, but we have the following **fail-stop property**: Let $[x;y,z]$ be valid, and let $[x;y',z']$ be verified such that $[y',z'] \neq [y,z]$. From $tt'^x = \sigma^y\tau^z = \sigma^{y+ez}$ and $tt'^x = \sigma^{y'}\tau^{z'} = \sigma^{y'+ez'}$ we get $y - y' = e(z'-z) \in \mathbb{Z}/q\mathbb{Z}$. Assume that $z' = z$, then we also have $y = y'$, thus $[y',z'] \neq [y,z]$, a contradiction. Hence we have $z' \neq z$ and $e = \frac{y-y'}{z'-z} \in \mathbb{Z}/q\mathbb{Z}$, thus Alice and Bob are able to compute the discrete logarithm $e = \log_\sigma(\tau)$, which we are assuming is difficult.

If $[x;y,z]$ and $[x';y',z']$ are valid with respect to the same key $[a,b,a',b']$, where $x' \neq x$, then $[a,b,a',b']$ can be determined; hence this is a one-time signature scheme: We have $y = a + xa'$ and $z = b + xb'$, and $y' = a + x'a'$ and $z' = b + x'b'$. Thus we are looking for the solutions in $(\mathbb{Z}/q\mathbb{Z})^4$ of the following system of linear equations, whose matrix has $(\mathbb{Z}/q\mathbb{Z})$-rank 4, and thus has a unique solution:

$$\begin{bmatrix} 1 & . & x & . \\ . & 1 & . & x \\ 1 & . & x' & . \\ . & 1 & . & x' \end{bmatrix} \cdot [a,b,a',b']^{\mathrm{tr}} = [y,z,y',z']^{\mathrm{tr}} \in (\mathbb{Z}/q\mathbb{Z})^4$$

b) Keys $[a,b,a',b'], [\widetilde{a},\widetilde{b},\widetilde{a}',\widetilde{b}'] \in (\mathbb{Z}/q\mathbb{Z})^4$ are called **equivalent** if $t := \sigma^a\tau^b = \sigma^{\widetilde{a}}\tau^{\widetilde{b}} \in \langle\sigma\rangle$ and $t' := \sigma^{a'}\tau^{b'} = \sigma^{\widetilde{a}'}\tau^{\widetilde{b}'} \in \langle\sigma\rangle$; hence the associated verification functions are identical. For fixed $b \in \mathbb{Z}/q\mathbb{Z}$ we have $\{\sigma^a\tau^b; a \in \mathbb{Z}/q\mathbb{Z}\} = \langle\sigma\rangle$, thus each element of $\langle\sigma\rangle$ arises from precisely q pairs $[a,b]$, hence each equivalence class of keys has precisely q^2 elements.

If $[x;y,z]$ is valid with respect to the key $[a,b,a',b']$, then it is valid with respect to precisely q of the keys equivalent to $[a,b,a',b']$: We are looking for simultaneous solutions of the equations $y = a + xa'$ and $z = b + xb'$, and $t = \sigma^a\tau^b = \sigma^{a+eb}$ and $t' = \sigma^{a'}\tau^{b'} = \sigma^{a'+eb'}$. Letting $c, c' \in \mathbb{Z}/q\mathbb{Z}$ such that $t = \sigma^c$ and $t' = \sigma^{c'}$, these are the solutions in $(\mathbb{Z}/q\mathbb{Z})^4$ of the following system of linear equations:

$$\begin{bmatrix} 1 & e & . & . \\ . & . & 1 & e \\ 1 & . & x & . \\ . & 1 & . & x \end{bmatrix} \cdot [a,b,a',b']^{\mathrm{tr}} = [c,c',y,z]^{\mathrm{tr}} \in (\mathbb{Z}/q\mathbb{Z})^4$$

Since the left hand matrix has $(\mathbb{Z}/q\mathbb{Z})$-rank 3, and the system by construction has a solution, we conclude that the set of solutions is an affine space of $(\mathbb{Z}/q\mathbb{Z})$-dimension 1, and thus there are precisely q solutions. $\sharp$

If $[x;y,z]$ is valid with respect to the key $[a,b,a',b']$, and $[x';y',z'] \in (\mathbb{Z}/q\mathbb{Z})^3$ such that $x' \neq x$, then there is at most one key equivalent to $[a,b,a',b']$ such that both $[x;y,z]$ and $[x';y',z']$ are valid with respect to that key; such a key exists only if $[x';y',z']$ is verified: We are looking for the solutions in $(\mathbb{Z}/q\mathbb{Z})^4$

of the following system of linear equations:

$$\begin{bmatrix} 1 & e & . & . \\ . & . & 1 & e \\ 1 & . & x & . \\ . & 1 & . & x \\ 1 & . & x' & . \\ . & 1 & . & x' \end{bmatrix} \cdot [a, b, a', b']^{\mathrm{tr}} = [c, c', y, z, y', z']^{\mathrm{tr}} \in (\mathbb{Z}/q\mathbb{Z})^6$$

Since the left hand matrix has $(\mathbb{Z}/q\mathbb{Z})$-rank 4, there is at most one solution. ♯

In conclusion, if $[x; y, z]$ is valid then it has been produced using one out of q keys in an equivalence class of keys. Given $x' \neq x$, for any forgery $[x'; y', z']$, i. e. $[x'; y', z']$ is verified, there is at most one key out of the above q ones such that $[x'; y', z']$ is valid with respect to that key. Hence the probability that $[x'; y', z']$ is valid is at most $\frac{1}{q}$. In particular, any Las-Vegas algorithm computing a selective forgery by a known-message attack has a success probability of at most $\frac{1}{q}$.

(22.4) The Chaum blind signature protocol [1982]. Blind signatures were invented for electronic payment systems. They are issued without Alice, the **signer**, knowing neither what is signed nor the signature. To achieve this, Bob, the **blinder**, blinds the message to be signed and unblinds the signature:

Let Alice have an RSA cryptosystem with public key $[n, e]$, with modulus $n \in \mathbb{N}$ and $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, and private key $d := e^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$. This is used as an RSA signature scheme, hence still the signature function is $g_e \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \colon x \mapsto x^d$, and the strong verification function $\gamma_e \colon (\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z}) \to \{0, 1\}$ is defined by $\gamma_e(x, y) := 1$ if and only if $y^e = x \in \mathbb{Z}/n\mathbb{Z}$.

Bob chooses $k \in (\mathbb{Z}/n\mathbb{Z})^*$ randomly, which he keeps private. If $x \in \mathbb{Z}/n\mathbb{Z}$ is to be signed, Bob sends the blinded message $x' := xk^e \in \mathbb{Z}/n\mathbb{Z}$ to Alice, who returns its RSA signature $y' := x'^d \in \mathbb{Z}/n\mathbb{Z}$ to Bob, who in turn unblinds it to find the signature $y := y'k^{-1} \in \mathbb{Z}/n\mathbb{Z}$; we indeed have $y = (xk^e)^d k^{-1} = x^d \in \mathbb{Z}/n\mathbb{Z}$.

Since $k^e \in \mathbb{Z}/n\mathbb{Z}$ is uniformly distributed, the blinded messages x' are as well, hence Alice does not obtain any information on the signed messages. Similarly, as $k^{-1} \in (\mathbb{Z}/n\mathbb{Z})^*$ is uniformly distributed, the blinded signatures y' are as well, and Alice does not obtain any information on the signatures either. If Alice keeps a track of all the blinded pairs she has produced and then gets hold of valid pair, again since $k \in (\mathbb{Z}/n\mathbb{Z})^*$ is uniformly distributed she is not able to associate it with a particular blinded pair; in particular, if Alice communicates with various blinders she is not able to identify the sender of a particular blinded message.

23 Identification

(23.1) Identification schemes. a) Identification is based on the idea that the **prover (claimant)** Alice has to convince the **verifier** Bob about her identity, by showing that she possesses some secret, at least with some high probability. The simplest identification schemes employ **weak identification (weak authentication)**, where Alice has to disclose her secret **password**, which subsequently is known to Bob or might be caught by an opponent.

The danger of **replay attacks**, i. e. an opponent could try to identify himself as being Alice by reusing a password, is remedied by **one-time password schemes**, where to avoid lists of passwords kept private by both Alice and Bob, e. g. the **Lamport one-time password scheme** proceeds as follows: Letting $\mathcal{X}$ be an alphabet and $f\colon \mathcal{X}^* \to \mathcal{X}^*$ be a one-way function, Alice and Bob agree on a common secret $w_0 \in \mathcal{X}^*$, and the password used for the i-th verification, where $i \in \mathbb{N}$, is $w_i := f(w_{i-1}) \in \mathcal{X}^*$. Still, this allows for a **pre-play attack**, hence Alice may present a password only if Bob is known to be authentic.

b) In **challenge-response identification (strong identification)** schemes the use of passwords is avoided: Bob sends a **challenge**, i. e. a question, to Alice, who returns as a **response** the answer to that question, having been computed using the secret without disclosing it.

E. g. cryptosystems $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ can be used to devise challenge-response identification schemes: If the cryptosystem is symmetric, Alice and Bob agree on a common private key $k \in \mathcal{K}$, where we assume $E_k D_k = \mathrm{id}_{\mathcal{P}}$, then Bob sends the challenge $x \in \mathcal{P}$ to Alice, who responds by returning $y := E_k(x) \in \mathcal{C}$ to Bob, who finally checks whether $D_k(y) = x \in \mathcal{P}$; still, both Alice and Bob know the secret, the private key k, in particular Bob is able to impersonate Alice.

If Alice and Bob have a public-key cryptosystem, let $e \in \mathcal{K}$ be the chosen public key and $d \in \mathcal{K}$ be a suitable private key, Bob chooses $x \in \mathcal{P}$, sends the challenge $y := E_e(x) \in \mathcal{C}$ to Alice, who returns $z := D_d(y) \in \mathcal{P}$ as a response to Bob, who checks whether $z = x \in \mathcal{P}$; only Alice knows of the secret, the private key d, but if an opponent is able to replace Alice's public key by his own one he is able to identify himself as Alice to Bob.

Challenge-response identification schemes often use **certification** by a **trusted authority**, who possesses a private signature function $g\colon \mathcal{X}^* \to \mathcal{X}^{\leq n}$, where $\mathcal{X}$ is an alphabet, with corresponding public verification function $\gamma\colon \mathcal{X}^* \times \mathcal{X}^{\leq n} \to \{0, 1\}$. Instead of simply publishing a certificate $v \in \mathcal{X}^*$, the prover Alice sends v to the trusted authority, who chooses an identification string $w \in \mathcal{X}^*$ identifying Alice, and publishes the **signed certificate** $[w, v, g(wv)]$. Since the signed certificate is publicly known, all the security objectives for signature schemes discussed earlier apply, to prevent attacks using a forged signed certificates.

(23.2) Proofs of knowledge. At best the protocol underlying a challenge-response identification scheme is a **proof of knowledge**, i. e. the protocol is **complete** inasmuch any valid identification is accepted, and **sound**, i. e. the existence of an oracle to forge identifications entails the existence of an expected polynomial time Monte-Carlo algorithm to compute the prover's secret. The protocol is **correct**, if forging identifications cannot be solved in expected

Monte-Carlo polynomial time; hence a sound protocol such that the prover's secret cannot be computed in expected Monte-Carlo polynomial time is correct.

Still, an opponent Oscar observing the execution of the protocol could collect information about Alice's secret. This leads to the following notion: A **transcript** of a protocol is the tuple of messages being sent on a single execution. Executing the protocol between a honest prover and a honest verifier induces a probability distribution on the set of transcripts. Then the protocol is called **(perfectly) zero-knowledge** if there is a **simulator**, i. e. a polynomial time Las-Vegas algorithm not interacting with the prover, producing transcripts with the same probability distribution as for honest interactions; the protocol is called **polynomially zero-knowledge** if there is a simulator producing tuples, not necessarily being transcripts, with a probability distribution which cannot be distinguished from the probability distribution for honest interactions in expected Monte-Carlo polynomial time.

This means that through pure observation of the execution of the protocol no further information about the secret can be obtained, hence only through active participation the verifier can be convinced that the prover indeed knows the secret. Still, by a **chosen-text attack**, i. e. a clever choice of challenges, a dishonest verifier might be able to discover at least part of the secret, hence the zero-knowledge property does not imply security of the identification scheme.

Typically, zero-knowledge proofs of knowledge are three-pass protocols being a combination of a **cut-and-choose** and a challenge-response protocol; the name refers to the method of sharing a cake by one person cutting and one choosing:

(23.3) The Fiat-Shamir identification scheme [1986]. Let $p \neq q \in \mathbb{N}$ be odd primes and let $n := pq \in \mathbb{N}$. Alice chooses a **key** $r \in (\mathbb{Z}/n\mathbb{Z})^*$ and lets the **certificate** be $s := r^2 \in (\mathbb{Z}/n\mathbb{Z})^*$, then the public key is $[n, s]$, while the private key is $[p, q, r]$. Identification runs in a three-pass protocol: Alice randomly chooses a **commitment** $t \in (\mathbb{Z}/n\mathbb{Z})^*$, and sends the **witness** $u := t^2 \in (\mathbb{Z}/n\mathbb{Z})^*$ to Bob; then Bob sends a **challenge** $x \in \{0, 1\}$, chosen independently from the witness, to Alice; then Alice returns the **response** $y := tr^x \in (\mathbb{Z}/n\mathbb{Z})^*$ to Bob, who finally checks whether $u = y^2 s^{-x} \in (\mathbb{Z}/n\mathbb{Z})^*$.

Since $y^2 s^{-x} = (tr^x)^2 r^{-2x} = t^2 = u \in (\mathbb{Z}/n\mathbb{Z})^*$ the protocol is complete. Let $[u, x, y] \in (\mathbb{Z}/n\mathbb{Z})^* \times \{0, 1\} \times (\mathbb{Z}/n\mathbb{Z})^*$ such that $u = y^2 s^{-x}$, then $u \in (\mathbb{Z}/n\mathbb{Z})^*$ is a square, thus there is $t \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $u = t^2$, hence $y^2 = (tr^x)^2$, thus choosing t appropriately amongst the four square roots of u, we conclude that $[u, x, y]$ is a witness-challenge-response triple, i. e. a transcript.

The commitment t must be chosen before Alice knows of the challenge x: Otherwise an opponent Oscar is able to impersonate Alice by choosing any $y \in (\mathbb{Z}/n\mathbb{Z})^*$, letting the witness be $u := y^2 s^{-x} \in (\mathbb{Z}/n\mathbb{Z})^*$, and answering the challenge x by the response y. Identification can be forged from the knowledge of the key r, breaking the identification scheme; in particular, if the same commitment t is used for both challenges $x \in \{0, 1\}$, then tr and t are known to Bob, hence $r^{\pm 1} \in (\mathbb{Z}/n\mathbb{Z})^*$, and thus from $r^2 = s$ the key r can be found.

Assume Oscar has an oracle to forge identifications, thus for some witness u Oscar is able to compute valid responses y_0 and y_1 to both challenges $x \in \{0,1\}$, respectively. Hence we have $s = su \cdot u^{-1} = (y_1 y_0^{-1})^2 \in (\mathbb{Z}/n\mathbb{Z})^*$, thus $y_1 y_0^{-1} \in (\mathbb{Z}/n\mathbb{Z})^*$ is one of the four possibilities for the key r, also being a square root of s. Hence Oscar is able to compute r in Las-Vegas polynomial time, thus the protocol is sound, hence a proof of knowledge of a square root r of the certificate s. Assuming that modular squaring is a cryptographic one-way function, i. e. computing modular square roots cannot be done in expected Las-Vegas polynomial time, as is conjectured (18.1), the protocol is correct.

We consider the transcripts $[u(t), x, y_r(t,x)] \in (\mathbb{Z}/n\mathbb{Z})^* \times \{0,1\} \times (\mathbb{Z}/n\mathbb{Z})^*$ as functions of $[t,x] \in (\mathbb{Z}/n\mathbb{Z})^* \times \{0,1\}$, considering $r \in (\mathbb{Z}/n\mathbb{Z})^*$ as a fixed parameter. The commitments t are chosen randomly, and t and the challenges x are chosen independently, hence the responses $y_r(t,x) := tr^x \in (\mathbb{Z}/n\mathbb{Z})^*$ are uniformly distributed, and x and $y_r(t,x)$ are independent. Then $u(t) = y_r(t,x)^2 s^{-x}$ is determined from x and $y_r(t,x)$, hence the set of transcripts is uniformly distributed, independently from the key r, and the uniform distribution can be simulated in polynomial time, thus the protocol is zero-knowledge. Still, no statement is made if Bob behaves dishonestly and chooses the challenges x depending on the witnesses u.

(23.4) The Guillou-Quisquater identification scheme [1988]. Let $p \neq q \in \mathbb{N}$ be primes, let $n := pq \in \mathbb{N}$, and let $e < n$ be a prime such that $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$. Alice chooses a key $r \in (\mathbb{Z}/n\mathbb{Z})^*$ and lets the certificate be $s := r^e \in (\mathbb{Z}/n\mathbb{Z})^*$, then the public key is $[n, s, e]$, while the private key is $[p, q, r]$. Identification runs in a three-pass protocol: Alice randomly chooses a commitment $t \in (\mathbb{Z}/n\mathbb{Z})^*$, and sends the witness $u := t^e \in (\mathbb{Z}/n\mathbb{Z})^*$ to Bob; then Bob sends a challenge $x \in \mathbb{Z}_e$, chosen independently from the witness, to Alice; then Alice returns the response $y := tr^x \in (\mathbb{Z}/n\mathbb{Z})^*$ to Bob, who finally checks whether $u = y^e s^{-x} \in (\mathbb{Z}/n\mathbb{Z})^*$.

Since $y^e s^{-x} = (tr^x)^e r^{-ex} = t^e = u \in (\mathbb{Z}/n\mathbb{Z})^*$ the protocol is complete. Let $[u, x, y] \in (\mathbb{Z}/n\mathbb{Z})^* \times \mathbb{Z}_e \times (\mathbb{Z}/n\mathbb{Z})^*$ such that $u = y^e s^{-x}$, then $u \in (\mathbb{Z}/n\mathbb{Z})^*$ is an e-th power, thus there is $t \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $u = t^e$, hence $y^e = (tr^x)^e$, letting $e' := e^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ yields $y = y^{ee'} = (tr^x)^{ee'} = tr^x \in (\mathbb{Z}/n\mathbb{Z})^*$, thus $[u, x, y]$ is a transcript. Identification can be forged from the knowledge of r.

Assume Oscar has an oracle to forge identifications, thus for some witness u Oscar is able to compute valid responses y and y' to challenges $x \neq x'$, respectively. Then we have $y^e s^{-x} = u = (y')^e s^{-x'} \in (\mathbb{Z}/n\mathbb{Z})^*$, thus $s^{x'-x} = (y'y^{-1})^e \in (\mathbb{Z}/n\mathbb{Z})^*$. Since $0 < |x' - x| < e$ and e is a prime, we have $\gcd(e, x'-x) = 1 = ae + b(x'-x)$ for some $a, b \in \mathbb{Z}$. Thus $s = s^{ae+(x'-x)b} = (s^a (y'y^{-1})^b)^e \in (\mathbb{Z}/n\mathbb{Z})^*$. Letting $e' := e^{-1} \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ yields $r = s^{e'} = s^a (y'y^{-1})^b \in (\mathbb{Z}/n\mathbb{Z})^*$. Hence Oscar is able to compute r in polynomial time, thus the protocol is sound, hence a proof of knowledge of the e-th root r of the certificate s. Assuming that modular powering is a cryptographic one-way function, i. e. computing modular e-th roots cannot be done in expected Las-Vegas polynomial time, as is conjectured, the protocol is correct.

We consider the transcripts $[u(t), x, y_r(t,x)] \in (\mathbb{Z}/n\mathbb{Z})^* \times \mathbb{Z}_e \times (\mathbb{Z}/n\mathbb{Z})^*$ as functions of $[t,x] \in (\mathbb{Z}/n\mathbb{Z})^* \times \mathbb{Z}_e$, considering $r \in (\mathbb{Z}/n\mathbb{Z})^*$ as a fixed parameter. The commitments t are chosen randomly, and t and the challenges x are chosen independently, hence the responses $y_r(t,x) := tr^x \in (\mathbb{Z}/n\mathbb{Z})^*$ are uniformly distributed, and x and $y_r(t,x)$ are independent. Then $u(t) = y_r(t,x)^e s^{-x}$ is determined from x and $y_r(t,x)$, hence the set of transcripts is uniformly distributed, independently from r, and the uniform distribution can be simulated in polynomial time, thus the protocol is zero-knowledge.

(23.5) The Schnorr identification scheme [1991]. Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, and let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity. Alice chooses a key $e \in \mathbb{Z}/q\mathbb{Z}$ and lets the certificate be $\sigma := \rho^e \in (\mathbb{Z}/p\mathbb{Z})^*$, then the public key is $[p, q, \rho, \sigma]$, while the private key is e. Identification runs in a three-pass protocol: Alice randomly chooses a commitment $f \in \mathbb{Z}/q\mathbb{Z}$, and sends the witness $\tau := \rho^f \in (\mathbb{Z}/p\mathbb{Z})^*$ to Bob; then Bob sends a challenge $x \in \mathbb{Z}/q\mathbb{Z}$, chosen independently from the witness, to Alice; then Alice returns the response $y := f - ex \in \mathbb{Z}/q\mathbb{Z}$ to Bob, who finally checks whether $\tau = \rho^y \sigma^x \in (\mathbb{Z}/p\mathbb{Z})^*$.

Since $\rho^y \sigma^x = \rho^{f-ex} \rho^{ex} = \tau \in (\mathbb{Z}/p\mathbb{Z})^*$, the protocol is complete. Let $[\tau, x, y] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ such that $\tau = \rho^y \sigma^x$, then letting $f := y + ex \in \mathbb{Z}/q\mathbb{Z}$ we have $\tau = \rho^f \in (\mathbb{Z}/p\mathbb{Z})^*$, thus $[\tau, x, y]$ is a transcript. Identification can be forged from the knowledge of the key e.

Assume Oscar has an oracle to forge identifications, thus for some witness τ Oscar is able to compute valid responses y and y' to challenges $x \neq x'$, respectively. Hence we have $\rho^y \sigma^x = t = \rho^{y'} \sigma^{x'} \in (\mathbb{Z}/p\mathbb{Z})^*$, thus $\rho^{y-y'} = \sigma^{e(x'-x)} \in (\mathbb{Z}/p\mathbb{Z})^*$, hence $y - y' = e(x'-x) \in \mathbb{Z}/q\mathbb{Z}$. Since $x' - x \in (\mathbb{Z}/q\mathbb{Z})^*$ we have $e = \frac{y-y'}{x'-x} \in (\mathbb{Z}/q\mathbb{Z})^*$. Hence Oscar is able to compute e in polynomial time, thus the protocol is sound, hence a proof of knowledge of the discrete logarithm $e = \log_\rho(\sigma)$ of the certificate σ. Assuming that modular exponentiation is a cryptographic one-way function, i. e. computing discrete logarithms cannot be done in expected Las-Vegas polynomial time, as is conjectured (18.1), the protocol is correct.

We consider the transcripts $[\tau(f), x, y_e(f,x)] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ as functions of $[f, x] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$, considering $e \in \mathbb{Z}/q\mathbb{Z}$ as a fixed parameter. The commitments f are chosen randomly, and f and the challenges x are chosen independently, hence the responses $y_e(f,x) := f - ex \in \mathbb{Z}/q\mathbb{Z}$ are uniformly distributed, and x and $y_e(f,x)$ are independent. Then $\tau(f) = \rho^{y_e(f,x)} \sigma^x$ is determined from x and $y_e(f,x)$, hence the set of transcripts is uniformly distributed, independently from e, and the uniform distribution can be simulated in polynomial time, thus the protocol is zero-knowledge.

(23.6) The Okamoto identification scheme [1993]. Let $q < p \in \mathbb{N}$ be primes such that $q \mid p-1$, let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive q-th root of unity, let $e \in (\mathbb{Z}/q\mathbb{Z})^*$ and $\sigma := \rho^e \in (\mathbb{Z}/p\mathbb{Z})^*$; hence σ also is a primitive q-th root of unity. The discrete logarithm $e = \log_\rho(\sigma)$ is only known to a trusted authority.

Then Alice randomly chooses a key $[a, b] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ and lets the certificate be $\tau := \rho^a \sigma^b \in (\mathbb{Z}/p\mathbb{Z})^*$, then the public key is $[p, q, \rho, \sigma, \tau]$, while the private key is $[a, b]$. Identification runs in a three-pass protocol: Alice randomly chooses a commitment $[c, d] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$, and sends the witness $u := \rho^c \sigma^d \in (\mathbb{Z}/p\mathbb{Z})^*$ to Bob; then Bob sends a challenge $x \in \mathbb{Z}/q\mathbb{Z}$, chosen independently from the witness, to Alice; then Alice returns the response $[y, z] := [c - ax, d - bx] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ to Bob, who finally checks whether $u = \rho^y \sigma^z \tau^x \in (\mathbb{Z}/p\mathbb{Z})^*$.

Since $\rho^y \sigma^z \tau^x = \rho^{c-ax} \sigma^{d-bx} \rho^{ax} \sigma^{bx} = \rho^c \sigma^d = u \in (\mathbb{Z}/p\mathbb{Z})^*$, the protocol is complete. Let $[u, x, y, z] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ such that $u = \rho^y \sigma^z \tau^x$, then letting $c := y + ax \in \mathbb{Z}/q\mathbb{Z}$ and $d := z + bx \in \mathbb{Z}/q\mathbb{Z}$ we get $\rho^c \sigma^d = \rho^{y+ax} \sigma^{z+bx} = \rho^y \sigma^z (\rho^a \sigma^b)^x = u$, thus $[u, x, y, z]$ is a transcript.

Let $\mathcal{K}_\tau := \{[a', b'] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}); \rho^{a'} \sigma^{b'} = \tau\}$ be the set of keys possibly used by Alice, hence we have $[a, b] \in \mathcal{K}_\tau$. For $[a', b'] \in \mathcal{K}_\tau$ we have $\rho^{a'-a} = \sigma^{b-b'}$, hence $e(b - b') = (a' - a) \in \mathbb{Z}/q\mathbb{Z}$, thus $\mathcal{K}_\tau = \{[a', b'] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}); a' = a - ek, b' = b + k, k \in \mathbb{Z}/q\mathbb{Z}\}$, hence $|\mathcal{K}_\tau| = q$. Let a transcript $[x, y, z]$ be fixed, and let $[a', b'] \in \mathcal{K}_\tau$, hence there is $k \in \mathbb{Z}/q\mathbb{Z}$ such that $a' = a - ek$ and $b' = b + k$. Hence letting $c' := c - ekx \in \mathbb{Z}/q\mathbb{Z}$ and $d' := d + kx \in \mathbb{Z}/q\mathbb{Z}$ yields $y = c - ax = (c' + ekx) - (a' + ek)x = c' - a'x$ and $z = d - bx = (d' - kx) - (b' - k)x = d' - b'x$, and $\rho^{c'} \sigma^{d'} = \rho^{c-ekx} \sigma^{d+kx} = \rho^c \sigma^d (\rho^{-e} \sigma)^{kx} = u$. Thus both the witness u and the response $[y, z]$ are also obtained using the key $[a', b']$ instead of $[a, b]$, and the commitment $[c', d']$ instead of $[c, d]$, hence identification can be forged from the knowledge of one of the keys in $\mathcal{K}_\tau$.

The protocol is a proof of knowledge of an element of $\mathcal{K}_\tau$, which is Alice's secret: Assume Oscar has an oracle to forge identifications, thus for some witness u Oscar is able to compute valid responses $[y, z]$ and $[y', z']$ to challenges $x \neq x'$, respectively. Hence we have $\rho^y \sigma^z \tau^x = u = \rho^{y'} \sigma^{z'} \tau^{x'}$, and thus $\rho^{y-y'} \sigma^{z-z'} = \tau^{x'-x}$. Hence letting $a' := \frac{y-y'}{x'-x} \in \mathbb{Z}/q\mathbb{Z}$ and $b' := \frac{z-z'}{x'-x} \in \mathbb{Z}/q\mathbb{Z}$ we get $\rho^{a'} \sigma^{b'} = \tau$ and thus $[a', b'] \in \mathcal{K}_\tau$. Hence Oscar is able to compute an element of $\mathcal{K}_\tau$ in polynomial time, and thus the protocol is sound.

For $[a', b'] \neq [a'', b''] \in \mathcal{K}_\tau$, since $b' = b''$ if and only if $a' = a''$, we have $b' \neq b''$, and thus from $\rho^{a''-a'} = \sigma^{b'-b''} = \rho^{e(b'-b'')}$ we conclude $e = \log_\rho(\sigma) = \frac{a''-a'}{b'-b''} \in \mathbb{Z}/q\mathbb{Z}$. Thus computing the discrete logarithm e polynomial time reduces to finding two distinct elements of $\mathcal{K}_\tau$. Still, Oscar's algorithm might first compute $[a', b'] \in \mathcal{K}_\tau$, using some challenge-response pair $[x, y, z]$ and some other pair with a different challenge, and then just return $y' := y - (x' - x)a' \in \mathbb{Z}/q\mathbb{Z}$ and $z' := z - (x' - x)b' \in \mathbb{Z}/q\mathbb{Z}$, for all $x' \neq x'' \in \mathbb{Z}/q\mathbb{Z}$. Then we have $\frac{y''-y'}{x'-x''} = \frac{-(x''-x)a' + (x'-x)a'}{x'-x''} = a' \in \mathbb{Z}/q\mathbb{Z}$ and $\frac{z''-z'}{x'-x''} = \frac{-(x''-x)b' + (x'-x)b'}{x'-x''} = b' \in \mathbb{Z}/q\mathbb{Z}$, hence Oscar might only obtain precisely one element of $\mathcal{K}_\tau$. But since e is only known to the trusted authority, who does not take part in the identification scheme, we may even allow Alice to collaborate. Then since $[a, b]$ has been chosen randomly, with probability $1-\frac{1}{q}$ we have $[a',b'] \neq [a,b]$, and hence this yields a polynomial time Las-Vegas algorithm to compute e. Hence if Oscar has an expected Las-Vegas polynomial time algorithm to forge identifications, there is an expected polynomial time Las-Vegas algorithm to compute e. Hence assuming that modular exponentiation is a cryptographic one-way function, i. e. computing discrete logarithms cannot be done in expected Las-Vegas polynomial time, as is conjectured (18.1), this shows that the protocol is correct.

We consider the transcripts $[u(c,d), x, y_{a,b}(c,d,x), z_{a,b}(c,d,x)] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ as functions of $[c,d,x] \in (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ considering $[a,b]$ as a fixed parameter. The commitments $[c,d]$ are chosen randomly, and $[c,d]$ and the challenges x are chosen independently, hence the responses $y_{a,b}(c,d,x) := c - ax \in \mathbb{Z}/q\mathbb{Z}$ and $z_{a,b}(c,d,x) := d - bx \in \mathbb{Z}/q\mathbb{Z}$ are uniformly distributed, and x and $[y,z]$ are independent. Then $u(c,d) = \rho^y \sigma^z \tau^x$ is determined from x and $[y,z]$, hence the set of transcripts is uniformly distributed, independently from the key $[a,b]$ used, and the uniform distribution can be simulated in polynomial time, thus the protocol is zero-knowledge.

(23.7) The Shamir permuted kernel identification scheme [1988]. Let $p \in \mathbb{N}$ be a prime, let $n \geq m \in \mathbb{N}$, and let $A := [a_{ij}]_{ij} \in \mathbb{F}_p^{n\times m}$ such that $\ker(A) := \{u \in \mathbb{F}_p^n; uA = 0\} \neq \{0\}$. For $\sigma \in \mathcal{S}_n$ and $u = [u_1, \ldots, u_n] \in \mathbb{F}_p^n$ let $u^\sigma := [u_{1\sigma^{-1}}, \ldots, u_{n\sigma^{-1}}] \in \mathbb{F}_p^n$ and $A^\sigma := [a_{i\sigma^{-1},j}]_{ij} \in \mathbb{F}_p^{n\times m}$; hence $[u^\sigma A^\sigma]_j = \sum_{i=1}^n u_{i\sigma^{-1}} a_{i\sigma^{-1},j} = \sum_{i=1}^n u_i a_{ij} = [uA]_j$ for $j \in \{1, \ldots, m\}$, thus $u^\sigma A^\sigma = uA$.

Let $\mathcal{X}$ be an alphabet, and choosing some suitable encoding let $\mathcal{D} := \{\pi \cdot v \in \mathcal{X}^*; \pi \in \mathcal{S}_n, v \in \mathbb{F}_p^n\} \subseteq \mathcal{X}^*$. Choosing $k \in \mathbb{N}$ large enough, let $h\colon \mathcal{D} \to \mathcal{X}^{\leq k}$ be an injective cryptographic one-way function; the elements of $\mathcal{D}$ have length in $O(n\ln(n) + n\ln(p))$, hence choosing $k \sim n\ln(n) + n\ln(p)$ suffices.

Alice chooses a key $\sigma \in \mathcal{S}_n$, and a certificate $u \in \mathbb{F}_p^n$ such that $u^\sigma \in \ker(A)$, then the public key is $[h, A, u]$, while the private key is σ. Identification runs in a three-pass protocol: Alice randomly chooses a commitment $[\tau, v] \in \mathcal{S}_n \times \mathbb{F}_p^n$, and sends the witness $[s,t] := [h(\tau \cdot vA), h(\sigma\tau \cdot v^\tau)] \in \mathcal{X}^{\leq k} \times \mathcal{X}^{\leq k}$ to Bob; then Bob sends a challenge $[\xi, x] \in \mathbb{F}_p \times \{0,1\}$, chosen independently from the witness, to Alice; then Alice returns the response $[w, \pi] := [(v + \xi u^\sigma)^\tau, \sigma^x \tau] \in \mathbb{F}_p^n \times \mathcal{S}_n$ to Bob, who finally checks whether $s := h(\tau \cdot vA) = h(\pi \cdot wA^\pi) \in \mathcal{X}^{\leq k}$ if $x = 0$, and $t := h(\sigma\tau \cdot v^\tau) = h(\pi \cdot (w - \xi u^\pi)) \in \mathcal{X}^{\leq k}$ if $x = 1$.

Since for $x = 0$ we have $\pi = \tau$ and $wA^\pi = (v + \xi u^\sigma)^\tau A^\tau = (v + \xi u^\sigma)A = vA$, implying $h(\pi \cdot wA^\pi) = h(\tau \cdot vA) = s$, while for $x = 1$ we have $\pi = \sigma\tau$ and $w - \xi u^\pi = (v + \xi u^\sigma)^\tau - \xi u^{\sigma\tau} = v^\tau$, implying $h(\pi \cdot (w - \xi u^\pi)) = h(\sigma\tau \cdot v^\tau) = t$, the protocol is complete.

Let $\mathcal{K}_u := \{\sigma' \in \mathcal{S}_n; u^{\sigma'} A = 0\}$ be the set of keys possibly used by Alice, hence we have $\sigma \in \mathcal{K}_u$, and identification can be forged from the knowledge of one of the keys in $\mathcal{K}_u$. If the commitment τ is used for both challenges $x \in \{0,1\}$, then from the responses $\sigma\tau \cdot \tau^{-1} = \sigma$ can be determined. If the commitment v is used for the challenges $[\xi, 0]$ and $[\xi', 0]$, where $\xi \neq \xi'$, we get the responses $[w, \tau]$ and $[w', \tau']$, thus $w^{\tau^{-1}} - \xi u^\sigma = v = w'^{\tau'^{-1}} - \xi' u^\sigma$ yields $u^\sigma = \frac{1}{\xi-\xi'}(w^{\tau^{-1}} - w'^{\tau'^{-1}}) \in \mathbb{F}_p^n$; some $\sigma' \in \mathcal{S}_n$ such that $u^{\sigma'} = \frac{1}{\xi-\xi'}(w^{\tau^{-1}} - w'^{\tau'^{-1}})$ can be found in polynomial time by sorting, and hence we have $\sigma' \in \mathcal{K}_u$.

Given $u \in \mathbb{F}_p^n$ and $A \in \mathbb{F}_p^{n\times m}$, finding a permutation $\sigma' \in \mathcal{S}_n$ such that $u^{\sigma'} \in \ker(A)$, if there exists any, is called the **permuted kernel problem**, i. e. the function problem associated to the relation $\mathcal{R} := \{[u, A; \sigma'] \in \mathbb{F}_p^n \times \mathbb{F}_p^{n\times m} \times \mathcal{S}_n; n \geq m \in \mathbb{N}, u^{\sigma'}A = 0\}$. Since the input lengths of $[u, A]$ and σ' are in $O(n^2\ln(p))$ and $O(n\ln(n))$, respectively, and computing $u^{\sigma'} \in \mathbb{F}_p^n$ and the product $u^{\sigma'}A \in \mathbb{F}_p^n$ need $O(n^2)$ and $O(n^2\ln(p)^2)$ bit operations, respectively, we conclude that $\mathcal{R}$ is a polynomial certificate for the following decision problem: Given $[u, A] \in \mathbb{F}_p^n \times \mathbb{F}_p^{n\times m}$, is there $\sigma' \in \mathcal{S}_n$ such that $u^{\sigma'}A = 0$? Hence the latter problem is in NP, and the associated function problem in FNP. The permuted kernel problem is NP-hard, see [10, Ch.10.6], hence it is conjectured that it cannot be solved in expected Las-Vegas polynomial time.

Assume Oscar has an oracle to forge identifications, thus for some witness $[s, t]$ Oscar is able to compute valid responses $[w_0, \pi_0]$ and $[w_1, \pi_1]$ to challenges $x \in \{0, 1\}$. Since h is injective we have $\pi_0 = \tau$ and $\pi_1 = \sigma\tau$, hence Oscar is able to compute σ in polynomial time, thus the protocol is sound, hence a proof of knowledge of σ. Assuming that the permuted kernel problem cannot be solved in expected Las-Vegas polynomial time, as is conjectured, the protocol is correct.

We consider the transcripts $[s(\tau, v), t_\sigma(\tau, v); \xi, x; w_\sigma(\tau, v, \xi), \pi_\sigma(\tau, x)] \in \mathcal{X}^{\leq k} \times \mathcal{X}^{\leq k} \times \mathbb{F}_p \times \{0, 1\} \times \mathbb{F}_p^n \times \mathcal{S}_n$ as functions of $[\tau, v; \xi, x] \in \mathcal{S}_n \times \mathbb{F}_p^n \times \mathbb{F}_p \times \{0, 1\}$ considering σ as a fixed parameter. The commitments $[\tau, v]$ are chosen randomly, and $[\tau, v]$ and the challenges $[\xi, x]$ are chosen independently, hence the responses $w_\sigma(\tau, v, \xi) := (v + \xi u^\sigma)^\tau \in \mathbb{F}_p^n$ and $\pi_\sigma(\tau, x) := \sigma^x\tau \in \mathcal{S}_n$ are uniformly distributed, and $[\xi, x]$ and $[w, \pi]$ are independent. Given $[\xi, x, w, \pi]$, from $\tau = \sigma^{-x}\pi$ and $v = w^{\pi^{-1}\sigma^x} - \xi u^\sigma$ the witnesses $s(\tau, v) := h(\tau \cdot vA) = h(\sigma^{-x}\pi \cdot wA^{\sigma^{-x}\pi})$ and $t_\sigma(\tau, v) := h(\sigma\tau \cdot v^\tau) = h(\sigma^{1-x}\pi \cdot (w - \xi u^{\sigma^{1-x}\pi}))$ are determined, thus the set of transcripts is uniformly distributed; in particular we recover the verification conditions using s for $x = 0$ and t for $x = 1$.

To simulate transcripts, values of h for arguments involving both π and $\sigma'\pi$, where $\pi \in \mathcal{S}_n$ and $\sigma' \in \mathcal{K}_u$, have to be computed, to do so an element $\sigma' \in \mathcal{K}_u$ has to be computed. Thus assuming that the latter cannot be done in expected Las-Vegas polynomial time, the protocol is not zero-knowledge. Instead, the uniform distribution on the set of tuples $[\widetilde{s}, \widetilde{t}; \xi, x, w, \pi]$ obtained from transcripts by letting $\widetilde{t} := h(\pi' \cdot w')$ for $x = 0$ and $\widetilde{s} := h(\pi' \cdot w')$ for $x = 1$, for some randomly chosen $\pi' \in \mathcal{S}_n$ and $w' \in \mathbb{F}_p^n$, can be simulated in polynomial time. A tuple $[\widetilde{s}, \widetilde{t}; \xi, x, w, \pi]$ is a transcript if and only if $\pi'\pi^{-1} \in \mathcal{K}_u$ for $x = 0$ and $\pi\pi'^{-1} \in \mathcal{K}_u$ for $x = 1$, where to decide this π' has to be computed as a preimage under h, which by assumption on h cannot be done in expected Monte-Carlo polynomial time. Hence the protocol still is polynomially zero-knowledge.

24 Interactive proof systems

(24.1) Interactive proof systems [Goldwasser-Micali-Rackoff, 1985]. **a)** Let $\mathcal{X}$ be an alphabet. An **interactive proof system** is a distributed algorithm, called a **protocol**, between a polynomial time non-deterministic Turing machine $\mathcal{B}$ and an exponential time deterministic Turing machine $\mathcal{A}$, hence in particular $\mathcal{A}$ might also be a polynomial time non-deterministic Turing machine. We proceed as follows: For $w \in \mathcal{X}^*$, beginning with $\mathcal{A}$, the machines alternatingly compute $a_i(w) \in \mathcal{X}^*$ and $b_i(w) \in \mathcal{X}^* \,\dot{\cup}\, \{\text{fail}\}$, for $i \in \mathbb{N}$, depending on $[w, a_1(w), b_1(w), , \ldots, b_{i-1}(w)]$ and $[w, a_1(w), b_1(w), , \ldots, b_{i-1}(w), a_i(w)]$, respectively, such that $l(a_i(w)), l(b_i(w)) \leq l(w)^k$ for some $k \in \mathbb{N}$, and terminating after at most $l(w)^k$ steps by $\mathcal{B}$ either accepting or rejecting w; hence $\mathcal{B}$ behaves both as a decision machine and a function machine. The finite tuple $[w, a_1(w), b_1(w), ...]$ is called a **transcript** of the interactive proof system.

b) A language $\mathcal{L} \subseteq \mathcal{X}^*$ is **decided** by an interactive proof system $[\mathcal{A}, \mathcal{B}]$, if there is $0 < \epsilon < 1$ such that $w \in \mathcal{L}$ is accepted in at least a fraction of $1-(1-\epsilon)^{l(w)}$ of the branches **(completeness)**, while $w \notin \mathcal{L}$ is accepted by any system $[\mathcal{A}', \mathcal{B}]$ in at most a fraction of $(1-\epsilon)^{l(w)}$ of the branches **(correctness)**. Let IP be the complexity class of languages being decided by an interactive proof system.

An interactive proof system deciding $\mathcal{L}$ is called **zero-knowledge**, if there is a polynomial time Las-Vegas algorithm, which for $w \in \mathcal{L}$ produces transcripts with the same probability distribution as is obtained by execution of the interactive proof system on input w.

c) Since $\mathsf{NP} \subseteq \mathsf{EXP}$, by letting $\mathcal{A}$ be a polynomial time non-deterministic decision machine and letting $\mathcal{B}$ just accept $\mathcal{A}$'s decision, we get $\mathsf{NP} \subseteq \mathsf{EXP} \subseteq \mathsf{IP}$. Letting $\mathcal{B}$ be a decision machine in BPP ignoring $\mathcal{A}$, we get $\mathsf{BPP} \subseteq \mathsf{IP}$.

By **Shamir's Theorem [1990]**, see [11, Thm.19.8], we have $\mathsf{IP} = \mathsf{PSPACE}$, where PSPACE is the complexity class of languages being decided by deterministic Turing machines needing space which is bounded polynomially in the input length. Since it is conjectured that $\mathsf{BPP} \not\subseteq \mathsf{NP} \neq \mathsf{P}$, it is also conjectured that IP is a much larger class than both NP and BPP.

(24.2) Graphs. **a)** A **(simple undirected) graph** $G := [V, E]$ is a finite set $V \neq \emptyset$ of **vertices** together with a set E of two-element subsets of V called **edges**; we may assume that $V = \{1, \ldots, n\}$ for some $n \in \mathbb{N}$. The symmetric group $\mathcal{S}_n$ acts on the set of two-element subsets of V, hence for a graph $G = [V, E]$ and $\pi \in \mathcal{S}_n$ we have the graph $G^\pi := [V, E^\pi]$. Graphs $G = [V, E]$ and $G' = [V, E']$ are called **isomorphic** if there is $\pi \in \mathcal{S}_n$ such that $E' = E^\pi$.

A graph is described by its edge set, where we have $|E| \leq \binom{n}{2} \leq n^2$, each edge consists of two numbers of input length in $O(\ln(n))$, thus the input length of a graph is in $O(n^2 \ln(n))$. Sorting an edge set E can be done by swapping adjacent entries of the list encoding E at most $O(|E|^2)$ times, hence needing at most $O(n^4)$ steps, being polynomial in the input length. Both applying $\pi \in \mathcal{S}_n$

to an edge set and comparing two edge sets need $O(n^2 \ln(n))$ steps.

b) We have the decision problem $\mathsf{GraphIsomorphism}$ (GRISO): Given graphs G and G' on the same vertex set, are they isomorphic? Hence the complementary decision problem is $\mathsf{GraphNonIsomorphism}$ (GRNISO): Given graphs G and G' on the same vertex set, are they non-isomorphic?

A non-deterministic algorithm deciding GRISO is given by running through all $\pi \in \mathcal{S}_n$ by successively choosing $1^\pi, 2^\pi, \ldots, n^\pi$, computing E^π, and comparing with E'. Since there are n choices to be made, this runs in polynomial time, thus GRISO is in NP; actually it is neither known whether GRISO is NP-complete nor whether it is in P, see [11, Ex.12.4].

By Stirling's Formula we have $n! \in O(e^{(n+\frac{1}{2})\ln(n)})$, thus there is a deterministic algorithm running in exponential time to decide GRNISO, thus GRNISO in EXP; GRNISO is neither known to be in NP nor in BPP, i. e. it is neither known whether GRISO is in coNP nor whether it is in BPP.

c) A **k-colouring** of a graph $G = [V, E]$, for $k \in \mathbb{N}$, is a map $\chi \colon V \mapsto \{1, \ldots, k\}$ such that for all $\{i, j\} \in E$ we have $\chi(i) \neq \chi(j)$. We have the decision problem $\mathsf{Graph\text{-}3\text{-}Colourability}$ ($\mathsf{GR3COL}$): Given a graph G, is there a 3-colouring of G?

A non-deterministic algorithm deciding GRISO is given by running through $v \in V$ and successively choosing $\chi(v) \in \{1, 2, 3\}$, and checking the colouring condition. Since there are n choices to be made and there are at most $\binom{n}{2}$ edges, this runs in polynomial time, thus $\mathsf{GR3COL}$ is in NP; actually $\mathsf{GR3COL}$ is NP-complete, see [11, Thm.9.8]. From that, since by (24.5) $\mathsf{GR3COL}$ is decided by a zero-knowledge interactive proof system, it can be deduced that all problems in NP are decided by a zero-knowledge interactive proof system [Goldreich-Micali-Wigderson, 1986], see [11, 12.3.6].

(24.3) Graph non-isomorphism. Let $G_1 = [V, E_1]$ and $G_2 = [V, E_2]$ be graphs, and $n := |V|$. For some fixed $k \geq 3$ and $i \in \{1, \ldots, n^k\}$ we proceed as follows: $\mathcal{B}$ randomly chooses $x \in \{1, 2\}$ and $\pi \in \mathcal{S}_n$, and sends $H := (G_x)^\pi$ to $\mathcal{A}$; then $\mathcal{A}$ computes $y \in \{1, 2\}$ such that G_y is isomorphic to H, choosing $y \in \{1, 2\}$ randomly if G_1 and G_2 are isomorphic, and sends y to $\mathcal{B}$; then $\mathcal{B}$ checks whether $y = x$. Finally, $\mathcal{B}$ accepts if and only if the verification succeeds for all $i \in \{1, \ldots, n^k\}$.

This is an interactive proof system for GRNISO: The lengths of the messages exchanged and the number of repetitions is bounded polynomially in the input length, and deciding GRISO as part of $\mathcal{A}$ runs in exponential time. If G_1 and G_2 are non-isomorphic, then we always have $y = x$, hence the protocol is complete. If G_1 and G_2 are isomorphic, then we have $y = x$ with probability $\frac{1}{2}$, independently from any choice of exponential time machine $\mathcal{A}'$ instead of $\mathcal{A}$, hence the probability that G_1 and G_2 are erroneously verified to be non-isomorphic is at most $(\frac{1}{2})^{n^k}$, thus the protocol is correct.

The protocol is zero-knowledge: The transcripts produced by execution of a round of the protocol, for a pair of non-isomorphic graphs, are of the form $[H, y]$, where H is a random permutation of a graph randomly chosen amongst G_1 and G_2, and $y \in \{1, 2\}$ is determined by $H \cong G_y$. Thus the transcripts are uniformly distributed. Hence by choosing $y \in \{1, 2\}$ and $\pi \in \mathcal{S}_n$ randomly and independently, and letting $H := (G_y)^\pi$, transcripts can be produced by a polynomial time Las-Vegas algorithm.

(24.4) Graph isomorphism. Let $G_1 = [V, E_1]$ and $G_2 = [V, E_2]$ be graphs, and $n := |V|$. For some fixed $k \geq 3$ and $i \in \{1, \ldots, n^k\}$ we proceed as follows: $\mathcal{A}$ randomly chooses $y \in \{1, 2\}$ and $\sigma \in \mathcal{S}_n$, and sends $H := (G_y)^\sigma$ to $\mathcal{B}$; then $\mathcal{B}$ chooses $x \in \{1, 2\}$, and sends x to $\mathcal{A}$; then $\mathcal{A}$ computes $\{\pi \in \mathcal{S}_n; (G_x)^\pi = H\}$ and randomly chooses one of its elements, which is chosen randomly if G_x and H are non-isomorphic, and sends π to $\mathcal{B}$; then $\mathcal{B}$ checks whether $(G_x)^\pi = H$. Finally, $\mathcal{B}$ accepts if and only if the verification succeeds for all $i \in \{1, \ldots, n^k\}$.

This is an interactive proof system for GRISO: The lengths of the messages exchanged and the number of repetitions is bounded polynomially in the input length, and computing $\{\pi \in \mathcal{S}_n; (G_x)^\pi = H\}$ runs in exponential time. If G_1 and G_2 are isomorphic, then we always have $(G_x)^\pi = H$, hence the protocol is complete. If G_1 and G_2 are non-isomorphic, then we have $(G_x)^\pi = H$ with probability $\frac{1}{2}$, independently from any choice of exponential time machine $\mathcal{A}'$ instead of $\mathcal{A}$, hence the probability that G_1 and G_2 are erroneously verified to be isomorphic is at most $(\frac{1}{2})^{n^k}$, thus the protocol is correct.

The protocol is zero-knowledge: The transcripts produced by execution of a round of the protocol, for a pair of isomorphic graphs, are of the form $[H, x, \pi]$, where H is a random permutation of a graph randomly chosen amongst G_1 and G_2, the element $x \in \{1, 2\}$ is randomly chosen, and π is randomly chosen in $\{\pi \in \mathcal{S}_n; (G_x)^\pi = H\}$. Thus the transcripts are uniformly distributed. Hence by choosing $x \in \{1, 2\}$ and $\pi \in \mathcal{S}_n$ randomly and independently, and letting $H := (G_x)^\pi$, transcripts can be produced by a polynomial time Las-Vegas algorithm.

(24.5) Graph 3-colourability. Let $G = [V, E]$ be a graph such that $V = \{1, \ldots, n\}$. We assume there is a Las-Vegas oracle computing RSA moduli. Then $\mathcal{A}$ first computes a 3-colouring $\chi\colon V \to \{0, 1, 2\}$, where if there is no 3-colouring $\mathcal{A}$ just chooses some map χ. For some fixed $k \geq 3$ and $l \in \{1, \ldots, n^k\}$ we then proceed as follows: $\mathcal{A}$ randomly chooses $\pi \in \mathcal{S}_{\{0,1,2\}}$, and for $i \in \{1, \ldots, n\}$ requests RSA moduli $n_i \leq n^k$ from the oracle and computes their factorisation $n_i = p_i q_i \in \mathbb{N}$, then $\mathcal{A}$ randomly chooses $e_i \in (\mathbb{Z}/\varphi(n_i)\mathbb{Z})^*$ and computes $d_i := e_i^{-1} \in (\mathbb{Z}/\varphi(n_i)\mathbb{Z})^*$, randomly chooses $y_i \in \{1, \ldots, \lfloor \frac{n_i}{3} \rfloor - 1\}$ and computes $z_i := (3y_i + \chi(i)^\pi)^{e_i} \in \mathbb{Z}/n_i\mathbb{Z}$, and sends $[n_i, e_i, z_i]$ for all $i \in \{1, \ldots, n\}$ to $\mathcal{B}$; then $\mathcal{B}$ randomly chooses an edge $\{i, j\} \in E$ and sends $[i, j]$ to $\mathcal{A}$; and $\mathcal{A}$ returns $[d_i, d_j]$ to $\mathcal{B}$; then $\mathcal{B}$ computes $x_i := z_i^{d_i} \in \mathbb{Z}_{n_i}$ and $x_j := z_j^{d_j} \in \mathbb{Z}_{n_j}$, and checks whether $x_i \not\equiv x_j \pmod 3$. Finally, $\mathcal{B}$ accepts if and only if the verification succeeds for all $l \in \{1, \ldots, n^k\}$.

This is an interactive proof system for $\mathsf{GR3COL}$: The lengths of the messages exchanged and the number of repetitions is bounded polynomially in the input length, and computing a 3-colouring and the factorisation of RSA moduli runs in exponential time. If there is a 3-colouring of G, then since $3y_i + \chi(i)^\pi \in \{3, \ldots, n_i - 3\}$ and $\chi(i)^\pi \in \{0, 1, 2\}$ we have $x_i = 3y_i + \chi(i)^\pi \in \mathbb{Z}_{n_i}$, and thus we always have $x_i \not\equiv x_j \pmod 3$, hence the protocol is complete. If there is no 3-colouring of G, then we have $x_i = x_j$ with probability at least $\frac{1}{|E|}$, independently from any choice of exponential time machine $\mathcal{A}'$ instead of $\mathcal{A}$, hence the probability that G is erroneously verified to have a 3-colouring is at most $(1 - \frac{1}{|E|})^{n^k}$, thus the protocol is correct.

The protocol is zero-knowledge: The transcripts produced by execution of a round of the protocol, for a graph having a 3-colouring, are of the form $[n_1, e_1, z_1, n_2, e_2, z_2, \ldots; i, j; d_i, d_j]$, where the n_s are produced by the oracle, the $e_s \in (\mathbb{Z}/p_s q_s \mathbb{Z})^*$ and $\{i, j\} \in E$ are chosen randomly and independently, and independent from the RSA moduli, thus determining d_i and d_j, and where the $z_s \in \mathbb{Z}/n_s\mathbb{Z}$ are chosen randomly and independently, and independent from the earlier choices, only subject to the condition $x_i \not\equiv x_j \pmod 3$, where $x_i := z_i^{d_i} \in \mathbb{Z}_{n_i}$. Thus the transcripts are uniformly distributed, and can be produced by a polynomial time Las-Vegas algorithm.

V Exercises

25 Exercises for Part I

(25.1) Exercise: Enumerating permutations.
a) Let $p_1, \ldots, p_d \in \mathbb{N}$. Show that any $x \in \{0, \ldots, (\prod_{j=1}^{d} p_j) - 1\}$ can be written uniquely in the form $x = \sum_{i=0}^{d-1} x_i \cdot \prod_{j=1}^{i} p_j$, where $x_i \in \{0, \ldots, p_{i+1} - 1\}$.
b) Let $n \in \mathbb{N}$. Using expansions with respect to $1, \ldots, n$ to give an explicit bijection $\{0, \ldots, n! - 1\} \rightarrow \mathcal{S}_{\{0,\ldots,n-1\}}$.

Hint for (b). Given $x = \sum_{i=0}^{n-1} x_i \cdot i!$, describe the associated permutation $\pi \in \mathcal{S}_{\{0,\ldots,n-1\}}$ by letting $(n-1)\pi := x_{n-1}$, and proceeding by induction, using that for any $j \in \{0, \ldots, n-1\}$ there is bijection $\{0, \ldots, n-2\} \rightarrow \{0, \ldots, n-1\} \setminus \{j\}$.

(25.2) Exercise: Sign map.
For $n \in \mathbb{N}_0$ let $\operatorname{sgn}\colon \mathcal{S}_n \rightarrow \{\pm 1\}\colon \pi \mapsto \prod_{1 \leq i < j \leq n} \frac{j\pi - i\pi}{j - i}$.

Show that $\operatorname{sgn}\colon \mathcal{S}_n \rightarrow \{\pm 1\}$ is well-defined, that is the formula on the right hand side indeed yields either 1 or -1, that it coincides with the sign map introduced in (2.1), and that it is a group homomorphism.

(25.3) Exercise: Encryption and decryption.
a) Implement programs to encode and decode words over $\mathcal{X}_{\text{latin}}$ to and from words over $\mathbb{Z}_{26}$; encrypting and decrypting words over $\mathbb{Z}_{26}$ using a shift cipher; launching a ciphertext-only attack against the shift cipher.
b) Decrypt the following text obtained by a shift cipher:

```
beeakfydjxuqyhyjiqryhtyjiqfbqduyjiikfuhcqd
```

Proof. b) See [12, Exc.1.5]. ♯

(25.4) Exercise: Shift cipher.
Give a formal definition of a cryptosystem where encryption is done as follows: For shift keys k and k', the letters of a word at odd and even positions are encrypted using k and k', respectively, and finally the resulting word is reversed.

Proof. See [2, Exc.3.16.2]. ♯

(25.5) Exercise: Substitution cipher.
Decrypt the following text obtained by a substitution cipher:

```
emglosudcgdncuswysfhnsfcykdpumlwgyicoxysipjck
qpkugkmgolicgincgacksnisacykzsckxecjckshysxcg
oidpkzcnkshicgiwygkkgkgoldsilkgoiusigledspwzu
gfzccndgyysfuszcnxeojncgyeoweupxezgacgnfglkns
acigoiyckxcjuciuzcfzccndgyysfeuekuzcsocfzccnc
iaczejncshfzejzegmxcyhcjumgkucy
```

Hint. `f` decrypts to `w`.

Proof. See [12, Exc.1.21]. ♯

(25.6) Exercise: Permutation cipher.
Decrypt the following text obtained by a permutation cipher of unknown block length:

`tgeemnelnntdroeoaahdoetcshaeirlm`

Proof. See [12, Exc.1.16]. ♯

(25.7) Exercise: Block ciphers.
a) Implement encryption and decryption for a block cipher over $\mathbb{Z}_2$ of block length $l \in \mathbb{N}$, running in one of the operation modes ECB, CBC, CFB, or OFB. In the former two modes assume that the block cipher in use is a permutation cipher, in the latter two modes assume that the auxiliary block cipher of block length $k \geq l$ is a permutation cipher.

b) Use this to encrypt the plaintext 101010101010, and to decrypt the ciphertext 111111111111, using $\pi := (1,2,3)$ and $l = 3$ in the former two modes, resp. $\pi := (1,2,3)$, initialisation vector 000 and $l = 2$ in the latter two modes.

(25.8) Exercise: Corrupted blocks.
We consider a block cipher over $\mathbb{Z}_2$, running in one of the operation modes ECB, CBC, CFB, or OFB. Let $[w_1, w_2, \ldots]$ be the sequence of ciphertext blocks transmitted. Which of the associated plaintext blocks $[v_1, v_2, \ldots]$ will be incorrectly decrypted in the following situations?

a) A ciphertext block w_i, for some $i \in \mathbb{N}$, is transmitted incorrectly, that is some block w_i' is received instead.

b) A ciphertext block w_i, for some $i \in \mathbb{N}$, is lost completely, that is the sequence received is $[w_1, w_2, \ldots, , w_{i-1}, w_{i+1}, \ldots]$.

(25.9) Exercise: Transposition cipher.
a) Give a formal definition of a cryptosystem where encryption is done as follows: Letting $m, n \in \mathbb{N}$, words of length mn are written row by row into an $(m \times n)$-matrix, then the rows are permuted, after that the columns are permuted, and finally words are read out column by column. How many keys are there?
b) Decrypt the following text obtained by such a **transposition cipher** of unknown block length:

`myamraruyiqtenctorahroywdsoyeouarrgdernogw`

Proof. See [12, Exc.1.26]. ♯

(25.10) Exercise: Involutory keys.
For a cryptosystem $[\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D}]$ such that $\mathcal{P}=\mathcal{C}$, a key $k\in\mathcal{K}$ such that $E_k^2=\mathrm{id}_{\mathcal{P}}$ is called **involutory**. Let $R:=\mathbb{Z}_n$, where $n\in\mathbb{N}$.
a) Determine the involutory keys of a Vigenère cipher of block length $l\in\mathbb{N}$.
b) Show that a key $[a,b]\in R^*\times R$ of an affine substitution cipher is involutory if and only if $a=a^{-1}\in R^*$ and $(a+1)b=0\in R$. Determine their number if n is a prime, and if $n=pq$ where $p,q\in\mathbb{N}$ are distinct primes.
c) Describe the involutory keys of a general substitution cipher over R. Determine their number for $n=26$.

Proof. See [12, Exc.1.6, 1.11]. ♯

(25.11) Exercise: Cramer's rule.
Let R be a commutative ring, let $A=[a_{ij}]\in\mathrm{GL}_n(R)$ and $b=[b_1,\ldots,b_n]\in R^n$.
a) Show that the system of R-linear equations $XA=b$, where $X=[X_1,\ldots,X_n]$ are indeterminates, has a unique solution $c=[c_1,\ldots,c_n]\in R^n$.
b) Use the adjointness theorem for determinants to prove **Cramer's rule**: Letting $a_i:=[a_{i1},\ldots,a_{in}]\in R^n$ be the rows of A, where $i\in\{1,\ldots,n\}$, we have

$$c_i=\frac{1}{\det(A)}\cdot\det(a_1,\ldots,a_{i-1},b,a_{i+1},\ldots,a_n)\in R.$$

(25.12) Exercise: Invertible matrices.
Let $n,m\in\mathbb{N}$. Show that $|\mathrm{GL}_n(\mathbb{Z}/m\mathbb{Z})|=m^{n^2}\cdot\prod_{p\text{ prime},p\,|\,m}\prod_{i=1}^{n}(1-\frac{1}{p^i})$.

(25.13) Exercise: Matrix inversion.
Check whether the following matrices over $\mathbb{Z}_{26}$ are invertible, and if so compute their inverses:

i) $\begin{bmatrix}7&23\\5&4\end{bmatrix}$ **ii)** $\begin{bmatrix}15&6&4\\14&19&22\\24&14&23\end{bmatrix}$

(25.14) Exercise: Bit permutations.
Let $n\in\mathbb{N}$.

a) Provide a permutation on $\mathbb{Z}_2^n$ which is not a bit permutation.

b) Provide a permutation on $\mathbb{Z}_2^n$ which is not affine linear.

(25.15) Exercise: Affine cipher.
Break an affine cipher of unknown block length using the following plaintext-ciphertext pair:

`adisplayedequation` ↦ `dsrmsioplxljbzullm`.

Proof. See [12, Exc.1.24]. ♯

(25.16) Exercise: Hill cipher.
Break a Hill cipher of unknown block length using the following plaintext-ciphertext pair:

$$\texttt{breathtaking} \quad \mapsto \quad \texttt{rupotentoifv}.$$

Proof. See [12, Exc.1.23]. ♯

(25.17) Exercise: Hill cipher.
Describe a ciphertext-only attack against a Hill cipher with block length 2, using the frequency of occurrence of pairs of letters in the plaintext language. Decrypt the following text obtained by such a cipher:

```
lmqetxyeagtxctuiewnctxlzewuaispzyvapewlmgqwya
xftcjmsqcadagtxlmdxnxsnpjqsyvapriqsmhnocvaxfv
```

Proof. See [12, Exc.1.25]. ♯

(25.18) Exercise: Vigenère cipher.
Decrypt the following text obtained by a Vigenère cipher of unknown block length:

```
kccpkbgufdphqtyavinrrtmvgrkdnbvfdetdgiltxrguddkotf
mbpvgegltgckqracqcwdnawcrxizakftlewrptycqkyvxchkft
poncqqrhjvajuwetmcmspkqdyhjvdahctrlsvskcgczqqdzxgs
frlswcwsjtbhafsiasprjahkjrjumvgkmitzhfpdispzlvlgwt
fplkkebdpgcebshctjrwxbafspezqnrwxcvycgaonwddkackaw
bbikftiovkcgghjvlnhiffsqesvyclacnvrwbbirepbbvfexos
cdygzwpfdtkfqiycwhjvlnhiqibtkhjvnpist
```

Proof. See [12, Exc.1.21]. ♯

(25.19) Exercise: Affine substitution cipher.
Decrypt the following text obtained by an affine substitution cipher:

```
kqerejebcppcjcrkieacuzbkrvpkrbcibqcarbjcvfcupkriof
kpacuzqepbkrxpeiieabdkpbcpfcdccafieabdkpbcpfeqpkaz
bkrhaibkapcciburccdkdccjcidfuixpafferbiczdfkabicbb
enefcupjcvkabpcydccdpkbcocperkivkscpicbrkijpkabi
```

Proof. See [12, Exc.1.21]. ♯

(25.20) Exercise: Decryption.
Decrypt the following text obtained by an unknown cipher:

```
bnvsnsihqceelsskkyerifjkxumbgykamqljtyavfbkvtdvbpv
vrjyylaokympqscgdlfsrllproygesebuualrwxmmasazlgled
fjbzavvpxwicgjxascbyehosnmulkceahtqokmflebkfxlrrfd
tzxciwbjsicbgawdvydhavfjxzibkcgjiweahttoewtuhkrqvv
rgzbxyiremmascspbnlhjmblrffjelhweylwistfvvyfjcmhyu
yrufsfmgesigrlwalswmnuhsimyyitccqpzsicehbccmzfegvj
yocdemmpghvaaumelcmoehvltipsuyilvgflmvwdvydbthfray
isysgkvsuuhyhggcktmblrx
```

Proof. See [12, Exc.1.21]. ♯

(25.21) Exercise: Autokey cipher.
Decrypt the following text obtained by an autokey cipher:

```
malvvmafbhbuqptsoxaltgvwwrg
```

Proof. See [12, Exc.1.28]. ♯

(25.22) Exercise: Vigenère stream cipher.
Given a Vigenère cipher over $R := \mathbb{Z}_n$, where $n \in \mathbb{N}$, of block length $l \in \mathbb{N}$, we obtain the synchronous **Vigenère stream cipher** using a seed key $b_0 \in R^l$ and the keystream $b_i := b_{i-1} + [1, \ldots, 1] \in R^l$, for $i \in \mathbb{N}$. What is its period?

Decrypt the following text obtained by a Vigenère stream cipher of unknown block length:

```
iymysilonrfncqxqjedshbuibcjuzbolfqyschatpeqgqjejng
nxzwhhgwfsukuljqaczkkjoaahgkemtafgmkvrdopxnehekznk
fskifrqvhhovxinphmrtjpywqgjwpuuvkfpoawpmrkkqzwlqdy
azdrmlpbjkjobwiwpsepvvqmbcryvcruzaaoumbchdagdiemsz
fzhaligkemjjfpciwkrmlmpinayofireaoldthitdvrmse
```

Proof. See [12, Exc.1.29]. ♯

(25.23) Exercise: Enigma cipher.
Given a substitution cipher with respect to the permutation $\pi \in \mathcal{S}_{\mathbb{Z}_{26}}$, we obtain a synchronous stream cipher using a seed key $k_0 \in \mathbb{Z}_{26}$ and the keystream $k_i := k_{i-1} + i \in \mathbb{Z}_{26}$, for $i \in \mathbb{N}$, where the i-th plaintext is encrypted by $\mathbb{Z}_{26} \to \mathbb{Z}_{26} \colon x \mapsto x\pi + k_i$. Let now $\pi \in \mathcal{S}_{\mathbb{Z}_{26}}$ be given as follows:

x	0	1	2	3	4	5	6	7	8	9	10	11	12
$x\pi$	23	13	24	0	7	15	14	6	25	16	22	1	19

x	13	14	15	16	17	18	19	20	21	22	23	24	25
$x\pi$	18	5	11	17	2	21	12	20	4	10	9	3	8

Decrypt the following text obtained by such an **Enigma cipher** with respect to the above permutation:

```
wswixggfctiorjlnlqmkbogxblcrjsrwsbmrktbxbhnnqfahxpdbkhuimjmt
```

Proof. See [12, Exc.1.30]. ♯

(25.24) Exercise: Linear recurrences.
Let $\mathbb{F}_2$ be the field with two elements. For all $d \in \{1, \ldots, 10\}$ determine the irreducible polynomials in $\mathbb{F}_2[X]$ of degree d. For all those polynomials and any seed in $\mathbb{F}_2^d$ determine the period of the associated linear recurrence.

(25.25) Exercise: Perfect security.
Let $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem, with probability distribution $\mu_{\mathcal{P}}$ and uniform distribution $\mu_{\mathcal{K}}$. Let the number $|\{k \in \mathcal{K}; E_k(x) = y\}|$ be independent of the particular choice of $x \in \mathcal{P}$ and $y \in \mathcal{C}$. Show that the cryptosystem is perfectly secure, and that $\mu_{\mathcal{C}}$ is the uniform distribution.

(25.26) Exercise: Perfect security of affine ciphers.
Let $R := \mathbb{Z}/n\mathbb{Z}$. Consider the affine cipher, the Hill cipher and the Vigenère cipher over R of block length $l \in \mathbb{N}$, and assume that the respective key sets are uniformly distributed. Which of these ciphers are perfectly secure?

Proof. See [12, Exc.2.3]. ♯

(25.27) Exercise: Latin square cryptosystems.
Let $n \in \mathbb{N}$. A **latin square** is a matrix $A \in \mathbb{Z}_n^{n \times n}$ such that any of its rows and columns contains every element of $\mathbb{Z}_n$. Given a latin square $A = [a_{ij}] \in \mathbb{Z}_n^{n \times n}$, let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_n$ and $\mathcal{E} := \{E_i; i \in \mathbb{Z}_n\}$, where $E_i \colon \mathbb{Z}_n \to \mathbb{Z}_n \colon j \mapsto a_{ij}$.
a) Show that this indeed defines a cryptosystem, called the associated **latin square cryptosystem**.
b) Provided $\mathcal{K}$ is uniformly distributed, show that it is perfectly secure.

Proof. See [12, Exc.2.2]. ♯

(25.28) Exercise: Entropy [Shannon, 1948].
Let X and Y be finite probability spaces.
a) Let $H(X) := -\sum_{x \in X; \mu_X(x) > 0} \mu_X(x) \cdot \log_2(\mu_X(x)) \in \mathbb{R}$ be the **entropy** of X. Show that $0 \leq H(X) \leq \log_2(|X|)$. When do we have $H(X) = 0$ or $H(X) = \log_2(|X|)$? How can entropy be interpreted as amount of information?
b) For $y \in Y$ such that $\mu_Y(y) > 0$ let $H(X|y) \in \mathbb{R}$ be the entropy of the conditional distribution $\mu_X(\cdot|y)$, and let $H(X|Y) := \sum_{y \in Y; \mu_Y(y) > 0} \mu_Y(y) \cdot H(X|y) \in$

$\mathbb{R}$ be the associated **conditional entropy**. Show that $H(X \times Y) = H(Y) + H(X|Y)$ and $H(X|Y) \leq H(X)$. When does equality hold? How can conditional entropy be interpreted?

Hint for a). Use the concavity of $\log_2 \colon \mathbb{R}_{>0} \to \mathbb{R}$ and the Jensen inequality.

Proof. See [12, Ch.2.4, 2.5]. $\sharp$

(25.29) Exercise: Entropy of cryptosystems.
Let $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem, together with independent probability distributions $\mu_{\mathcal{P}}$ and $\mu_{\mathcal{K}}$, and associated distribution $\mu_{\mathcal{C}}$. We keep the notation of (25.28).
a) Show that $H(\mathcal{K}|\mathcal{C}) = H(\mathcal{K}) + H(\mathcal{P}) - H(\mathcal{C})$ and $H(\mathcal{K}|\mathcal{C}) \geq H(\mathcal{P}|\mathcal{C})$. How can the latter inequality be interpreted?
b) Show that $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ is perfectly secure if and only if $H(\mathcal{P}|\mathcal{C}) = H(\mathcal{P})$.

Proof. See [12, Ch.2.6, Exc.2.11, 2.12]. $\sharp$

26 Exercises for Part II

(26.1) Exercise: Artin's Theorem.
Let U be a finite subgroup of the multiplicative group $\mathbb{C}^*$. Show that U consists of roots of unity, and is cyclic.

(26.2) Exercise: RSA block cipher.
a) Implement programs to encrypt and decrypt texts using an **RSA block cipher**: Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$ be the modulus, and let $l := \lfloor \log_{26} n \rfloor \in \mathbb{N}$ be the block length. Words in $\mathcal{X}_{\text{latin}}^l$ are first encoded into $\mathbb{Z}_{26}^l$, and then via 26-adic expansion considered as elements of $\mathbb{Z}_{26^l} \subseteq \mathbb{Z}_n$.

Try to implement encryption and decryption as efficient as possible. How can the ring isomorphism $\mathbb{Z}/n\mathbb{Z} \to (\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/q\mathbb{Z})$ be used to optimise decryption?

b) Decrypt the following text obtained using $n := 18721$ and $e := 43$:

$$13130, 95, 7342, 13805, 8347, 10022, 5164, 13434, 18716, 13434, 14498$$

c) Describe a protocol failure if contrary to the above we just let $l := 1$, and decrypt the following text obtained using $n := 18721$ and $e := 25$:

$$365, 0, 4845, 14930, 2608, 2608, 0$$

Proof. a) See [12, Exc.5.12, 5.13] and [2, Ch.8.3.9]. **c)** See [12, Exc.5.15]. $\sharp$

(26.3) Exercise: Protocol failure of RSA block ciphers.
a) Implement programs launching cycling attacks, low exponent attacks and common modulus attacks against an RSA block cipher.
b) Encrypting the plaintext from (26.2)(b) using $n := 18721$ and $e := 25$ yields the following ciphertext:

$$17980, 294, 9866, 6834, 14416, 5476, 3744, 10541, 16721, 10541, 4508.$$

Decrypt this text using a cycling attack and using a common modulus attack.

(26.4) Exercise: Generalised cycling attacks.
Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, and let $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$. What happens in a cycling attack against an RSA cryptosystem, if for $x \in \mathbb{Z}/n\mathbb{Z}$ the minimal $k \in \mathbb{N}$ such that $\gcd(x^{e^k} - x, n) > 1$ is searched for? What does this mean for the security of RSA cryptosystems?

Proof. See [10, Ch.8.2]. ♯

(26.5) Exercise: Low decryption exponent attack.
a) Implement a program launching a low decryption exponent attack against an RSA cryptosystem.
b) For which of the admissible encryption keys for the modulus $n := 18721$ is this attack successful?
c) Factor the modulus $n := 317940011$ using $e := 77537081$.

Proof. See [12, Exc.5.32]. ♯

(26.6) Exercise: Fixed points.
Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, and let $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$. Show that $|\{x \in \mathbb{Z}/n\mathbb{Z}; x^e = x\}| = (1 + \mathrm{ggT}(e-1, p-1)) \cdot (1 + \mathrm{ggT}(e-1, q-1))$. What does this mean for RSA cryptosystems?

Proof. See [12, Exc.5.18]. ♯

(26.7) Exercise: Perfect security of the RSA cryptosystem.
Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, and let $\mathcal{K} := (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ be uniformly distributed. Which of the RSA cryptosystems with $\mathcal{P} = \mathcal{C} = \mathbb{Z}/n\mathbb{Z}$ and $\mathcal{P} = \mathcal{C} = (\mathbb{Z}/n\mathbb{Z})^*$, respectively, are perfectly secure?

(26.8) Exercise: Semantic security of the RSA cryptosystem.
Let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, and let $e, d \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ such that $E_e D_d = \mathrm{id}_{\mathbb{Z}_n}$. Let $\sigma_e \colon \mathbb{Z}_n \to \mathbb{Z}_2 \colon y \mapsto D_d(y) \pmod 2$, and let $\tau_e \colon \mathbb{Z}_n \to \mathbb{Z}_2$ such that $\tau_e(y) = 0$ if and only if $D_d(y) \in \{0, \ldots, \frac{n-1}{2}\}$. Show that computing σ_e, computing τ_e, and computing D_d are polynomial time equivalent.

Proof. See [12, Ch.5.9.1, Exc.5.34]. $\sharp$

(26.9) Exercise: Legendre and Jacobi symbols.
a) Let $p \in \mathbb{N}$ be an odd prime. Show that $Q_p := \{x^2 \in (\mathbb{Z}/p\mathbb{Z})^*; x \in (\mathbb{Z}/p\mathbb{Z})^*\} \leq (\mathbb{Z}/p\mathbb{Z})^*$ is the unique subgroup of index 2. The elements of Q_p are called **quadratic residues**, and those of $N_p := (\mathbb{Z}/p\mathbb{Z})^* \setminus Q_p$ are called **quadratic non-residues**.
b) Let the **Legendre symbol** $\left(\frac{\cdot}{p}\right) : \mathbb{Z}/p\mathbb{Z} \to \{0, 1, -1\}$ be defined by

$$\left(\frac{x}{p}\right) := \begin{cases} 0, & \text{if } x = 0, \\ 1, & \text{if } x \in Q_p, \\ -1, & \text{if } x \in N_p. \end{cases}$$

Show that $\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right)$ for all $x, y \in \mathbb{Z}/p\mathbb{Z}$.
c) Now let $p \neq q \in \mathbb{N}$ be odd primes, let $n := pq \in \mathbb{N}$, and let the **Jacobi symbol** be defined by $\left(\frac{\cdot}{n}\right) := \left(\frac{\cdot}{p}\right)\left(\frac{\cdot}{q}\right) : \mathbb{Z}/n\mathbb{Z} \to \{0, 1, -1\}$. Letting $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$, show that $\left(\frac{x}{n}\right) = \left(\frac{E_e(x)}{n}\right)$. What does this mean for the RSA cryptosystem?

Proof. See [12, Ch.5.4]. $\sharp$

(26.10) Exercise: Choice of moduli.
Show that an RSA cryptosystem with modulus $n := 2189284635403183$ can be broken by elementary arithmetic.

Proof. See [12, Exc.5.29]. $\sharp$

(26.11) Exercise: RSA factorisation.
a) Implement the Las-Vegas algorithm to factor an RSA modulus using a known pair of encryption-decryption keys.
b) Factor the modulus $n := 18721$ using the pair $e := 43$ and $d := 9859$. How many factorisation witnesses are there in $\mathbb{Z}/n\mathbb{Z}$?

(26.12) Exercise: Square roots.
Let $p \in \mathbb{N}$ be a prime such that $p \equiv 5 \pmod 8$ and let $x \in (\mathbb{Z}/p\mathbb{Z})^*$.
a) Let $y := x^2 \in (\mathbb{Z}/p\mathbb{Z})^*$ and $z := y^{\frac{p-1}{4}} \in (\mathbb{Z}/p\mathbb{Z})^*$. Show that $z \in \{\pm 1\}$.
b) If $z = 1$, show that $x \in \{\pm y^{\frac{p+3}{8}}\}$. If $z = -1$, show that $x \in \{\pm 2y \cdot (4y)^{\frac{p-5}{8}}\}$.

Hint for (b). Use the fact that $2 \in N_p$, see (26.9).

Proof. See [12, Exc.6.21] and [10, Alg.3.37]. $\sharp$

(26.13) Exercise: Rabin block cipher.
a) Implement programs to encrypt and decrypt texts using a **Rabin block cipher** for primes $p \neq q \in \mathbb{N}$ such that $p, q \equiv 3 \pmod 4$ and modulus $n := pq \in \mathbb{N}$, which is defined as described in (26.2).

Decrypt the following text obtained using $n := 458933$, by first factoring n:

23178	14674	231736	113930	93071	353028	75628
5276	194098	144990	452983	1878	358306	304746
352950	435566	6494	124866	2862	174345	23021
142072	16254	146018	96674	30028	81000	333981

b) Implement the Las-Vegas algorithm to factor the Rabin modulus using a square root oracle for $(\mathbb{Z}/n\mathbb{Z})^*$. Use it to factor the modulus $n := 458933$. How many factorisation witnesses are there in $\mathbb{Z}/n\mathbb{Z}$?

(26.14) Exercise: ElGamal block cipher.
Implement programs to solve the discrete logarithm problem and to encrypt and decrypt texts using a randomised **ElGamal block cipher**, which is defined as described in (26.2), but paying attention to the fact that $0 \notin \mathbb{Z}_p^*$.

Decrypt the following text obtained using $p = 17579$, $\rho = 2$ and $\alpha = 16295$, by solving a discrete logarithm problem:

$[4112, 14926]$	$[14877, 1629]$	$[13851, 10582]$	$[830, 15113]$
$[11462, 3281]$	$[16556, 12332]$	$[1696, 9016]$	$[16695, 5440]$
$[10011, 6489]$	$[13746, 13450]$	$[14241, 17132]$	$[364, 16819]$
$[4048, 10319]$	$[14754, 17568]$	$[8249, 13863]$	$[14776, 13274]$

(26.15) Exercise: ElGamal cryptosystem.
a) Let $p \in \mathbb{N}$ be a prime, let $\mu \in \mathbb{F}_p[X]$ be irreducible of degree $n \in \mathbb{N}$, and let $K := \mathbb{F}_p[X]/\mu\mathbb{F}_p[X]$ be the field with p^n elements. Implement programs for addition, multiplication, inversion, and exponentiation in K, as well as finding primitive roots and solving the discrete logarithm problem in K^*.

b) Specify an ElGamal cryptosystem over K^*, where $|K| \geq 27$, and implement programs to encrypt and decrypt texts using a randomised ElGamal cipher, where letters in $\mathcal{X}_{\text{latin}}$ are encoded into $\mathbb{Z}_{26}$, and K^* is enumerated lexicographically, e. g. for $p = n = 3$ we let $K^* = [1, 2, X, X{+}1, X{+}2, 2X, \ldots, 2X^2{+}2X{+}2]$.

c) Let $p := 3$ and $n := 3$ as well as $\mu := X^3 - X + 1 \in \mathbb{F}_3[X]$. Show that μ is irreducible and that $X \in K := \mathbb{F}_3[X]/\mu\mathbb{F}_3[X]$ is a primitive root of K^*. Decrypt the following text obtained using $\rho := X$ and unknown key $\alpha \in K^*$:

$$\begin{array}{ll}
[-X^2+X-1, X-1] & [-X^2+1, X^2] \\
[X, X^2] & [X, X^2-1] \\
[-X^2+1, X] & [X^2+X-1, X] \\
[-X^2+X, -X^2-1] & [-X^2+1, -X^2-X-1] \\
[-X^2+X, -X^2-X-1] & [-X^2+X-1, -X] \\
[X+1, -X^2+1] & [-X^2+X-1, -X^2+X-1] \\
[X, X^2+X] & [X^2+1, 1] \\
[-X^2+1, -X^2+X] & [-X^2-X-1, X^2-1] \\
[-X^2, -X^2+X] & [-X^2+X, X] \\
[X^2+X-1, -X^2+X+1] & [-X^2-X, -X^2-X-1] \\
[-X^2-X-1, X-1] & [X^2-X-1, -X]
\end{array}$$

Proof. See [12, Exc.6.11, 6.12]. $\sharp$

(26.16) Exercise: Semantic security of the ElGamal cryptosystem.
Let $p \in \mathbb{N}$ be an odd prime, let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root, and let $\alpha \in (\mathbb{Z}/p\mathbb{Z})^*$ be a public key of an ElGamal cryptosystem.
a) Show **Euler's criterion**: For $x \in (\mathbb{Z}/p\mathbb{Z})^*$ we have $x \in Q_p$, see (26.9), if and only if $x^{\frac{p-1}{2}} = 1 \in (\mathbb{Z}/p\mathbb{Z})^*$.
b) Given a ciphertext $[\beta, y] \in (\mathbb{Z}/p\mathbb{Z})^* \times (\mathbb{Z}/p\mathbb{Z})^*$, show that the Legendre symbol $\left(\frac{x}{p}\right)$ of the associated plaintext $x \in (\mathbb{Z}/p\mathbb{Z})^*$ can be computed in polynomial time. What does this mean for the security of the ElGamal cryptosystem?

Proof. See [12, Ch.6.7.2]. $\sharp$

(26.17) Exercise: Discrete logarithms.
Let $p \in \mathbb{N}$ be a prime such that $p \equiv 3 \pmod 4$, let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root, for $x \in (\mathbb{Z}/p\mathbb{Z})^*$ let $\log_\rho(x) = \sum_{i \geq 0} l_i(x) \cdot 2^i \in \mathbb{Z}_{p-1}$, where $l_i(x) \in \{0, 1\}$.
a) Show that for any $x \in (\mathbb{Z}/p\mathbb{Z})^*$ we have $l_0(x) \neq l_0(-x)$, and that $x \in Q_p$, see (26.9), if and only if $l_0(x) = 0$.
b) Assume we have an l_1-oracle for $x \in (\mathbb{Z}/p\mathbb{Z})^*$. Show that $\log_\rho(x) \in \mathbb{Z}_{p-1}$ can be computed in polynomial time.

Proof. See [12, Ch.6.7.1]. $\sharp$

(26.18) Exercise: Discrete logarithm factorisation.
Let $p \neq q \in \mathbb{N}$ be odd primes, and let $n := pq \in \mathbb{N}$.
a) Show that for $x \in (\mathbb{Z}/n\mathbb{Z})^*$ we have $|x| \mid \operatorname{lcm}(p-1, q-1)$, and that there is $\rho \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $|\rho| = \operatorname{lcm}(p-1, q-1)$.
b) Let $\rho \in (\mathbb{Z}/n\mathbb{Z})^*$ be as above. Show how a discrete logarithm oracle in $\langle \rho \rangle$ can be used to factor n.

Proof. See [12, Exc.6.7]. $\sharp$

(26.19) Exercise: Chor-Rivest cryptosystem.
a) Let $n, k \in \mathbb{N}$ such that $k \leq n$. Implement a program to convert $x \in \mathbb{Z}_{\binom{n}{k}}$ to and from a sequence $[x_1, \ldots, x_n] \in \{0, 1\}^n$, where precisely k of the x_i are equal to 1, such that $x = \sum_{i=1}^{n} x_i \cdot \binom{n-i}{k_i}$, where $0 \leq k_i \leq k$.
b) Implement programs to encrypt and decrypt texts using the Chor-Rivest cryptosystem for a prime $p \in \mathbb{N}$ and an irreducible polynomial $\mu \in \mathbb{F}_p[X]$ of degree $d \geq 2$, such that $X \in \mathbb{F}_p[X]/\mu\mathbb{F}_p[X]$ is a primitive root.

27 Exercises for Part III

(27.1) Exercise: Turing machines.
Give the definition of a Turing machine over $\mathcal{X} := \{0, 1\}$
a) which for $w \in \mathcal{X}^*$ outputs $0w \in \mathcal{X}^*$,
b) which for $n \in \mathbb{N}_0$ in binary representation outputs the successor $n + 1 \in \mathbb{N}$.

Proof. **a)** See [11, Ex.2.1]. **b)** See [11, Ex.2.2]. ♯

(27.2) Exercise: Multiple string Turing machines.
a) Generalising the notion of a Turing machine, give a formal definition of a k**-string Turing machine**, where $k \in \mathbb{N}$, having transition function

$$\tau \colon \left(\mathcal{X} \,\dot{\cup}\, \mathcal{Y}\right)^k \times (\mathcal{S} \setminus \{s_\infty\}) \longrightarrow \left((\mathcal{X} \,\dot{\cup}\, \mathcal{Y}) \times \{\leftarrow, \uparrow, \rightarrow\}\right)^k \times \mathcal{S},$$

and acting on a suitable set of configurations, where inputs are provided on the string 1, and outputs are provided on string k.
b) Show that a language $\mathcal{L}$ which is accepted (decided) by a k-string Turing machine running in time f, is accepted (decided) by a conventional Turing machine running in time $O(f^2)$.

Proof. See [11, Ch.2.3] and [1, La.10.1]. ♯

(27.3) Exercise: Chernoff bound.
Let $X := \{0, 1\}$ be a probability space such that $0 < \mu(1) = \epsilon < 1$. Let $x_1, \ldots, x_k \in X$ be independent choices, for $k \in \mathbb{N}$, and let $x := \sum_{i=1}^{k} x_i$. Show that for $0 < \lambda \leq 1$ we have $\mu(x \geq (1 + \lambda)\epsilon k) \leq e^{-\frac{\lambda^2 \epsilon k}{2}}$.

Hint. Consider e^{tx} for $t \in \mathbb{R}$, and use the convexity of the exponential function, and $\mu(z \geq sE(z)) \leq \frac{1}{s}$ for $s > 0$, where $E(z)$ denotes the expectation value of the random variable z.

Proof. See [11, La.11.9]. ♯

(27.4) Exercise: Asymptotic behaviour.
a) Give an elementary proof that $\ln(n!) \in O(n \ln n)$ and $n \ln n \in O(\ln(n!))$.
b) Show that for $k \in \mathbb{N}$ we have $\sum_{i=1}^{n} i^k \sim \frac{n^{k+1}}{k+1}$ for $n \to \infty$.
c) For $n \in \mathbb{N}$ let $f_n \in \mathbb{N}$ denote the n-th **Fibonacci number** be given by $f_1 = f_2 = 1$, and $f_n := f_{n-1} + f_{n-2}$ for $n \geq 3$. Find a function $g \colon \mathbb{N} \to \mathbb{R}_{>0}$ such that $f_n \sim g(n)$.
d) Show that for the following functions $\mathbb{N} \to \mathbb{R}_{>0}$ we have $f \in o(g)$ whenever $f < g$, where $0 < \epsilon < 1 < c$:

$$1 < \ln\ln(n) < \ln(n) < e^{(\ln(n))^{\frac{1}{2}} (\ln\ln(n))^{\frac{1}{2}}} < n^{\epsilon} < n^{c} < n^{\ln(n)} < c^{n} < n^{n} < c^{c^{n}}.$$

Proof. **a)** See [8, Ex.2.2.2, Exc.2.2.4]. **b)** See [8, Ex.2.1.3].
c) See [8, Exc.2.2.2]. **d)** See [10, Ex.2.58]. ♯

(27.5) Exercise: Time estimates.
a) Let $n \in \mathbb{N}$. Show that $n!$ can be computed using $O(n^2 \ln(n)^2)$ bit operations, and give an estimate of the number of bit operations needed to compute n^n.
b) By induction show that $\sum_{i=1}^{n} i^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$, for $n \in \mathbb{N}$. Give estimates of the number of bit operations needed to compute the left hand side and the right hand of this equation.
c) For $n \in \mathbb{N}$ let $f_n \in \mathbb{N}$ denote the n-th Fibonacci number. Give estimates of the number of bit operations needed to compute $\sum_{i=1}^{n} f_i$ and $\prod_{i=1}^{n} f_i$.
d) For $1 \neq z \in \mathbb{N}$ and $n \in \mathbb{N}$ let $P_{z,n} := \{p \in \mathbb{N}; b_z(p) \leq n, p \text{ prime}\}$. Give estimates of the number of bit operations needed to compute $\sum P_{z,n}$ and $\prod P_{z,n}$.

Proof. **a)** See [8, Ex.2.3.3] and [8, Exc.2.3.1]. **b)** See [8, Exc.2.3.3].
c) See [8, Exc.2.3.5]. **d)** See [8, Exc.2.3.6]. ♯

(27.6) Exercise: Euclidean algorithm.
Let $q, m, n \in \mathbb{N}$ such that $q \neq 1$. Show that $\gcd(q^m - 1, q^n - 1) = q^{\gcd(m,n)} - 1$.

(27.7) Exercise: Lamé's Theorem.
Show that the Euclidean algorithm needs at most $\lfloor \log_{\frac{1+\sqrt{5}}{2}}(\sqrt{5} \cdot n) \rfloor - 2$ division steps to compute a greatest common divisor of $m, n \in \mathbb{N}$, where $m \geq n$.

Proof. See [7, Ch.3.3] and [3, Thm.1.3.2]. ♯

(27.8) Exercise: Primitive roots.
Implement a program to compute the minimal primitive root in $\mathbb{Z}_p^*$, where $p \in \mathbb{N}$ is a prime, and compute those for $p < 10^3$. What can be observed?

(27.9) Exercise: Divisibility.
a) Determine all $n \in \mathbb{N}$ such that $n \mid 2^n - 1$.
b) Determine all $n \in \mathbb{N}$ such that $n \mid x^{n+1} - x$ for all $x \in \mathbb{Z}_n$.

Proof. **a)** See [3, Exc.8.9.5]. **b)** See [3, Exc.8.9.4]. ♯

(27.10) Exercise: Primality tests.
a) Show **Wilson's Theorem**: For $1 \neq n \in \mathbb{N}$ we have $(n-1)! \equiv -1 \pmod{n}$ if and only if n is a prime.
b) Determine the number of bit operations needed for the primality tests based on trial division and on Wilson's Theorem, respectively.

Proof. **b)** See [11, Exc.10.4.8] and (27.5). ♯

(27.11) Exercise: Pocklington test.
a) Let $1 \neq n \in \mathbb{N}$, and let $p \in \mathbb{N}$ be a prime such that $p^a \mid n-1$ but $p^{a+1} \not| \, n-1$ for some $a \in \mathbb{N}$. Assume there is $x \in \mathbb{Z}_n$ such that $x^{n-1} = 1 \in \mathbb{Z}/n\mathbb{Z}$ and $\gcd(x^{\frac{n-1}{p}} - 1, n) = 1$. Show that for any $d \in \mathbb{N}$ dividing n we have $p^a \mid d-1$.
b) Let $m, l \in \mathbb{N}$ such that $n-1 = ml$, where $\gcd(m,l) = 1$ and $m > \sqrt{n}$. Show that n is a prime if and only if for any prime $p \in \mathbb{N}$ dividing m there is $x_p \in \mathbb{Z}_n$ fulfilling the conditions in a). How can this be used as a primality test?

Proof. **a)** See [3, Prop.8.3.1]. **b)** See [3, Cor.8.3.2]. ♯

(27.12) Exercise: Fermat test.
a) Implement a program performing the Fermat compositeness test, and determine the fraction of Fermat witnesses for the composite integers $n \leq 10^4$.
b) Which of the integers $10^{200} + 349$ and $10^{200} + 357$ are composite?

Proof. See [7, Exc.18.3]. ♯

(27.13) Exercise: Fermat witnesses.
Let $n \in \mathbb{N}$ be composite, and let $x \in (\mathbb{Z}/n\mathbb{Z})^*$ be Fermat witness such that $\gcd(x^{n-1} - 1, n) > 1$. Show that n is not a prime power.

Proof. See [7, Exc.18.6]. ♯

(27.14) Exercise: Fermat liars.
a) Let $p \in \mathbb{N}$ be a prime such that $2p-1$ is a prime as well, and let $n := p(2p-1)$. Show that a fraction of $\frac{1}{2}$ of the elements of $(\mathbb{Z}/n\mathbb{Z})^*$ are Fermat liars.
b) Let $p \neq q \in \mathbb{N}$ be primes such that $p, q \equiv 3 \pmod{4}$ and $\gcd(p-1, q-1) = 2$, and let $n := pq \in \mathbb{N}$. Show that $\{x^{n-1} \in (\mathbb{Z}/p\mathbb{Z})^*; x \in (\mathbb{Z}/p\mathbb{Z})^*\} \leq (\mathbb{Z}/p\mathbb{Z})^*$ has index 2, and use this to deduce the fraction of Fermat liars in $(\mathbb{Z}/n\mathbb{Z})^*$.

Proof. **a)** See [7, Exc.18.5]. **b)** See [7, Exc.18.4]. ♯

(27.15) Exercise: Carmichael numbers.
Let $1 \neq n \in \mathbb{N}$ be composite.
a) Show that n is a Carmichael number, if and only if n is squarefree and for any prime divisor $p \in \mathbb{N}$ of n we have $p - 1 \mid n - 1$.
b) Show that if n is a Carmichael number, then n is odd and has at least three different prime divisors.
c) Given an odd prime $p \in \mathbb{N}$ show that there are only finitely many Carmichael numbers n being divisible by p and having precisely three different prime divisors. Determine those for $p = 3$ and $p = 5$.
d) Implement a program to compute the Carmichael numbers $n \leq 10^4$.

Proof. See [7, Exc.18.9, 18.10]. ♯

(27.16) Exercise: Carmichael function.
Let $1 \neq n = \prod_{i=1}^{r} p_i^{a_i} \in \mathbb{N}$ be odd, where the p_i are pairwise distinct primes and $a_i \in \mathbb{N}$. Let $\lambda(n) := \mathrm{lcm}(\varphi(p_1^{a_1}), \ldots, \varphi(p_r^{a_r})) \in \mathbb{N}$ be the **Carmichael function**.
a) For $x \in (\mathbb{Z}/n\mathbb{Z})^*$ show that $x^{\lambda(n)} = 1$. Use this to show that n is a Carmichael number if and only if $\lambda(n) \mid n - 1$.
b) Show that $V_n := \{x \in (\mathbb{Z}/n\mathbb{Z})^*; x^{\frac{\lambda(n)}{2}} = \pm 1\} \leq (\mathbb{Z}/n\mathbb{Z})^*$ is a subgroup. Show that $V_n = (\mathbb{Z}/n\mathbb{Z})^*$ if and only if $r = 1$, i. e. n is a prime power. Use this to device a primality test for n, whenever n is known not to be a perfect power.

Proof. See [7, Exc.18.13, 18.14]. ♯

(27.17) Exercise: Strong pseudoprimes.
a) Show that there are composite odd $9 \neq n \in \mathbb{N}$ such that a fraction of $\frac{1}{4}$ of the elements of $(\mathbb{Z}/n\mathbb{Z})^*$ are strong liars.
b) For $t \in \mathbb{N}$ let $n_t := \prod\{p \in \{3, \ldots, t\}; p \text{ prime}\} \in \mathbb{N}$. Determine the strong liars for n_t.

Proof. a) See [10, Ex.4.22]. **b)** See [10, Ex.4.27]. ♯

(27.18) Exercise: Miller-Rabin test.
a) Implement a program performing the Miller-Rabin compositeness test, and determine the fraction of strong witnesses for the composite integers $n \leq 10^4$.
b) Is 1 195 068 768 795 265 792 518 361 315 725 116 351 898 245 581 composite? In this case try to determine its factorisation.

Proof. b) See [3, Ch.8.2]. ♯

(27.19) Exercise: Factorisation.
a) Implement the Lucas primality test, the ρ factorisation method and the $p-1$ factorisation method, and compute the factorisation of 15 770 708 441.
b) Show that the Fermat numbers F_{10}, F_{11} and F_{12} have at least two, two, and three prime divisors $< 10^9$, respectively. Which of the cofactors are prime?

Proof. **a)** See [12, Ex.5.9]. **b)** See [7, Ch.19.1]. $\sharp$

28 Exercises for Part IV

(28.1) Exercise: Computing square roots.
Let p be an odd prime. Let $\mathsf{QuadraticResidue}$ be the following decision problem, see (26.9): Given $x \in (\mathbb{Z}/p\mathbb{Z})^*$, is $x \in Q_p$? Let $\mathsf{SquareRoot}$ be the following function problem: Given $x \in (\mathbb{Z}/p\mathbb{Z})^*$, if possible find $y \in (\mathbb{Z}/p\mathbb{Z})^*$ such that $y^2 = x$.
a) Show that $\mathsf{SquareRoot}$ is the function problem associated to the decision problem $\mathsf{QuadraticResidue}$.
b) Show that $\mathsf{QuadraticResidue}$ can be decided in running time $O(\ln(p)^3)$.
c) Show that $\mathsf{SquareRoot}$ can be solved by a Las-Vegas algorithm with error bound $\frac{1}{2}$ having running time $O(\ln(p)^4)$.
d) Let $p \not\equiv 1 \pmod{8}$. Show that $\mathsf{SquareRoot}$ can be solved in time $O(\ln(p)^3)$.

Proof. **c)** See [8, Ch.6.1.8]. **d)** See [10, Ch.3.5.1]. $\sharp$

(28.2) Exercise: Collisions.
Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} \subseteq \mathcal{X}^*$ be finite, let $h\colon \mathcal{D} \rightarrow \mathcal{X}^{\leq n}$ be a compression function, where $s := |\mathcal{D}|$ and $t := |\mathcal{X}^{\leq n}|$, for $v \in \mathcal{X}^{\leq n}$ let $s_v := |h^{-1}(v)|$, and let $\sigma := \frac{1}{t} \cdot \sum_{v \in \mathcal{X}^{\leq n}} s_v$.
a) Let $c := |\{\{w, w'\} \subseteq \mathcal{D}; w \neq w', h(w) = h(w')\}|$. Show that $c = \frac{1}{2} \cdot (\sum_{v \in \mathcal{X}^{\leq n}} s_v^2) - \frac{s}{2}$ and $\sum_{v \in \mathcal{X}^{\leq n}} (s_v - \sigma)^2 = 2c + s - \frac{s^2}{t}$, and conclude that $c \geq \frac{1}{2} \cdot (\frac{s^2}{t} - s)$, with equality if and only if $s_v = \frac{s}{t}$ for all $v \in \mathcal{X}^{\leq n}$.
b) Let $c' := |\{[w, w'] \in \mathcal{D} \times \mathcal{D}; h(w) = h(w')\}|$. Show that $c' \geq \frac{s}{t}$, with equality if and only if $s_v = \frac{s}{t}$ for all $v \in \mathcal{X}^{\leq n}$.

Proof. See [12, Exc.4.1, 4.2]. $\sharp$

(28.3) Exercise: Preimages and second preimages.
Let $\mathcal{X}$ be an alphabet, let $\mathcal{D} \subseteq \mathcal{X}^*$ be finite, let $h\colon \mathcal{D} \rightarrow \mathcal{X}^{\leq n}$ be a compression function, where $s := |\mathcal{D}|$ and $t := |\mathcal{X}^{\leq n}|$, and for $v \in \mathcal{X}^{\leq n}$ let $s_v := |h^{-1}(v)|$. Assume the random oracle model, and that the oracle is queried $k \in \mathbb{N}$ times.
a) Show that the success probability to find a preimage of $v \in \mathcal{X}^{\leq n}$ is given as $1 - \frac{\binom{s-s_v}{k}}{\binom{s}{k}}$, that the expected success probability over $\mathcal{X}^{\leq n}$ is given as $\epsilon_k := 1 - \frac{1}{t} \cdot \sum_{v \in \mathcal{X}^{\leq n}} \frac{\binom{s-s_v}{k}}{\binom{s}{k}}$, and determine ϵ_1.
b) Show that the success probability to find a second preimage of $w \in \mathcal{D}$ is given as $1 - \frac{\binom{s-s_v}{k}}{\binom{s-1}{k}}$, that the expected success probability over $\mathcal{D}$ is given as $\delta_k := 1 - \frac{1}{s} \cdot \sum_{v \in \mathcal{X}^{\leq n}} \frac{s_v \cdot \binom{s-s_v}{k}}{\binom{s-1}{k}}$, and determine δ_1.

Proof. See [12, Exc.4.3, 4.4]. $\sharp$

(28.4) Exercise: Iterated functions.
Let $\mathcal{X} := \{0, 1\}$ and let $h\colon \mathcal{X}^{2n} \to \mathcal{X}^n$ be a compression function where $n \in \mathbb{N}$. Let $g\colon \mathcal{X}^{4n} \to \mathcal{X}^n$ be defined as follows: Writing $w \in \mathcal{X}^{2n}$ as $w = w'w''$ where $w', w'' \in \mathcal{X}^n$, let $g(w) := h(h(w')h(w''))$. Show that finding a collision for h reduces to finding a collision for g. How many queries of h are necessary?

Proof. See [12, Exc.4.9]. $\sharp$

(28.5) Exercise: Forgeries.
Let $\mathcal{X} := \{0, 1\}$, let $\mathcal{C} := \mathcal{X}^m$ where $m \in \mathbb{N}$, and let $[\mathcal{C}, \mathcal{C}, \mathcal{H}, \mathcal{E}, \mathcal{D}]$ be a cryptosystem where $\mathcal{H} \subseteq \mathcal{X}^*$. Let $n \in \mathbb{N}$ and let $h\colon \mathcal{H} \times \mathcal{C}^n \to \mathcal{C}$ be the keyed compression function defined by $h_u(w_1, \ldots, w_n) := \bigoplus_{i=1}^n E_u(w_i)$, where $E_u \in \mathcal{E}$ and $\oplus$ denotes the 'exclusive or' operation on bit strings.
a) Show that an existential forgery can be computed using one query of h.
b) Given $[w_1, \ldots, w_n] \in \mathcal{C}^n$, show that a selective forgery $[w_1, \ldots, w_n; w'] \in \mathcal{C}^n \times \mathcal{C}$ can be computed using two queries of h.

Proof. See [12, Exc.4.12]. $\sharp$

(28.6) Exercise: Deception probabilities.
Determine the deception probabilities $\epsilon_{0,1}$ for the keyed compression function $h\colon \{1, \ldots, 6\} \times \{1, \ldots, 4\} \to \{1, \ldots, 3\}$ given by the **authentication matrix**

key	1	2	3	4
1	1	1	2	3
2	1	2	3	1
3	2	1	3	1
4	2	3	1	2
5	3	2	1	3
6	3	3	2	1

Proof. See [12, Exc.4.15]. $\sharp$

(28.7) Exercise: Strongly universal functions.
Let $p \in \mathbb{N}$ be a prime and let $R := \mathbb{Z}/p\mathbb{Z}$.
a) Show that the functions $h'\colon R \times R \times R \to R\colon [a, b; x] \mapsto h'_{a,b}(x) := ax + b$ and $h''\colon R \times R \times R \to R\colon [a, b; x] \mapsto h''_{a,b}(x) := (x + a)^2 + b$ are strongly universal.
b) Let $l \in \mathbb{N}$ and $\mathcal{D} := \{0, 1\}^l \setminus \{[0, \ldots, 0]\}$. Show that the function $h\colon R^l \times \mathcal{D} \to R\colon [a_1, \ldots, a_l; x_1, \ldots, x_l] \mapsto h_{a_1,\ldots,a_l}(x_1, \ldots, x_l) := \sum_{i=1}^l a_i x_i$ is strongly universal.

Proof. a) See [12, Thm.4.12] and [12, Exc.4.16]. **b)** See [12, Thm.4.13]. $\sharp$

(28.8) Exercise: Higher deception probabilities.
Let $h\colon \mathcal{H} \times \mathcal{D} \to \mathcal{X}^{\leq n}$ be a strongly universal keyed hash function. Show that there is a Las-Vegas algorithm finding a forgery, using at most two queries of h, having success probability $\epsilon \geq \frac{|\mathrm{im}(h)|^2}{|\mathcal{H}|}$.

Proof. See [12, Exc.4.14]. ♯

(28.9) Exercise: Strongly k-universal functions.
Let $\mathcal{X}$ be an alphabet, let $\mathcal{H}, \mathcal{D} \subseteq \mathcal{X}^*$ be finite such that $|\mathcal{D}| \geq k \geq 1$ and let $h\colon \mathcal{H} \times \mathcal{D} \to \mathcal{X}^{\leq n}$ be a keyed compression function, where $n \in \mathbb{N}$. Then h is called **strongly k-universal**, if for $w_1, \ldots, w_k \in \mathcal{D}$ pairwise distinct and $v_1, \ldots, v_k \in \mathrm{im}(h)$ we have $|\{u \in \mathcal{H}; h_u(w_i) = v_i \text{ for } i \in \{1, \ldots, k\}\}| = \frac{|\mathcal{H}|}{|\mathrm{im}(h)|^k}$.
a) Show that strong k-universality, for $k \geq 2$, implies strong $(k-1)$-universality.
b) Let $p \in \mathbb{N}$ be a prime, let $R := \mathbb{Z}/p\mathbb{Z}$, and let $k \in \mathbb{N}$. Show that $h\colon R^k \times R \to R\colon [a_0, \ldots, a_{k-1}; x] \mapsto h_{a_0,\ldots,a_{k-1}}(x) := \sum_{i=0}^{k-1} a_i x^i$ is strongly k-universal.

Proof. See [12, Exc.4.17]. ♯

(28.10) Exercise: ElGamal signature scheme.
a) Implement the signature and verification functions of the ElGamal signature scheme, and various types of attacks.
b) Let $p := 31847$ and $\rho := 5$ and $\alpha := 26379$. Show that $[20543; 20679, 11082]$ is valid, and determine the keys by solving a single discrete logarithm problem.
c) Let $p := 31847$ and $\rho := 5$ and $\alpha := 25703$. Show that $[8990; 23972, 31396]$ and $[31415; 23972, 20481]$ are valid, and determine the keys without solving a discrete logarithm problem.
d) Let $p := 467$ and $\rho := 2$ and $\alpha := 132$. Show that $[100; 29, 51]$ is valid, and compute forgeries using a key-only attack and a known-message attack.

Proof. See [12, Exc.7.1, 7.2, 7.4]. ♯

(28.11) Exercise: ElGamal signature scheme with predictable keys.
a) Let $[p, \rho, \alpha]$ be the fixed public key of an ElGamal signature scheme. For $i \in \{1, 2\}$ let $b_i \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $b_2 - b_1 = 2$, and let $[x_i; \beta_i, y_i] \in \mathbb{Z}_{p-1} \times \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$ be valid, where $\beta_i = \rho^{b_i} \in (\mathbb{Z}/p\mathbb{Z})^*$. Show that the ElGamal signature scheme can be broken without solving a discrete logarithm problem.
b) Let $p := 28703$ and $\rho := 5$ and $\alpha := 11339$. Show that $[12000; 26530, 19862]$ and $[24567; 3081, 7604]$ are valid, and break this scheme using the above attack.

Proof. See [12, Exc.7.3]. ♯

(28.12) Exercise: Schnorr signature scheme.
Show that the Schnorr signature scheme is vulnerable to a key-reuse attack.

Proof. See [12, Exc.7.8]. $\sharp$

(28.13) Exercise: Lamport signature scheme.
Let two elements in $\{0,1\}^k$ be signed using the Lamport signature scheme with the same key. How many forgeries can be computed by a known-message attack?

Proof. See [12, Exc.7.14]. $\sharp$

(28.14) Exercise: Zero-knowledge proofs.
Let $p \in \mathbb{N}$ be a prime, let $\rho \in (\mathbb{Z}/p\mathbb{Z})^*$ be a primitive root, let $e \in \mathbb{Z}/(p-1)\mathbb{Z}$ and let $\sigma := \rho^e \in (\mathbb{Z}/p\mathbb{Z})^*$. Give a zero-knowledge proof of knowledge of the discrete logarithm $e = \log_\rho(\sigma)$. Is this protocol correct?

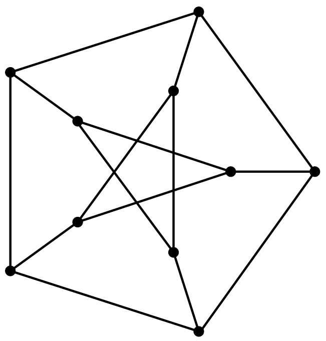
Proof. See [2, Exc.14.4.1]. $\sharp$

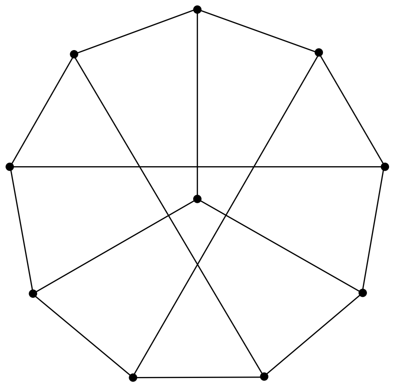
(28.15) Exercise: Feige-Fiat-Shamir identification scheme.
Give a generalisation of the Fiat-Shamir identification scheme, where the private key is a tuple $[r_1, \ldots, r_k]$ of $k \in \mathbb{N}$ independently chosen $r_i \in (\mathbb{Z}/n\mathbb{Z})^*$, and the challenges are tuples $[x_1, \ldots, x_k]$ of k independently chosen $x_i \in \{0,1\}$. Show that this is a zero-knowledge proof of knowledge of a square root of the certificate $s \in (\mathbb{Z}/n\mathbb{Z})^*$. What is its advantage compared to the original protocol?

Proof. See [2, Exc.14.4.3] and [10, Ch.10.4.2]. $\sharp$

(28.16) Exercise: Matrix kernels.
Let $p \in \mathbb{N}$ be a prime, let $n, m \in \mathbb{N}$, and let $A \in \mathbb{F}_p^{n \times m}$ be randomly chosen.
a) For $n \leq m$ determine the probability that $\ker(A) := \{u \in \mathbb{F}_p^n; uA = 0\} = \{0\}$.
b) For $n \in \{m, m+1\}$ determine the probability that $\dim_F(\ker(A)) = 1$.

(28.17) Exercise: Graph isomorphism and non-isomorphism.
Implement the interactive proof systems for graph isomorphism and graph non-isomorphism, and apply them to the following graphs:





(28.18) Exercise: Mental poker.
Let $n \in \mathbb{N}$. Describe a protocol between Alice and Bob producing uniformly distributed pairs $[a, b]$ such that $a \neq b \in \{1, \ldots, n\}$, where a is only known to Alice and b is only known to Bob.

Proof. See [11, Ch.12.2]. ♯

(28.19) Exercise: Quadratic residues and non-residues.
Let $p \neq q \in \mathbb{N}$ be odd primes, and let $n := pq \in \mathbb{N}$.
a) Let QuadraticResidue be the following decision problem: Given $x \in (\mathbb{Z}/n\mathbb{Z})^*$, is $x \in Q_n := \{y^2 \in (\mathbb{Z}/n\mathbb{Z})^*; y \in (\mathbb{Z}/n\mathbb{Z})^*\}$? Give an interactive proof system deciding QuadraticResidue. Is the protocol zero-knowledge?
b) Let QuadraticNonResidue be the complementary decision problem: Given $x \in (\mathbb{Z}/n\mathbb{Z})^*$, is $x \in N_n := (\mathbb{Z}/n\mathbb{Z})^* \setminus Q_n$? Give an interactive proof system deciding QuadraticNonResidue. Is the protocol zero-knowledge?

Proof. **a)** See [13, Ch.13.2]. **b)** See [13, Exc.13.1]. ♯

(28.20) Exercise: Coin flipping via telephone.
Which objectives should a protocol between Alice and Bob fulfil, who are only connected by telephone and want to make a decision by coin flipping? Describe such a protocol.

Hint. Let $p \neq q \in \mathbb{N}$ be odd primes and $n := pq \in \mathbb{N}$, and use the difficulty of the quadratic residuosity problem.

Proof. See [13, Ch.13.3]. ♯

(28.21) Exercise: Subgroup membership.
Let SubgroupMembership be the following decision problem: Given $n \in \mathbb{N}$ and $x, y \in (\mathbb{Z}/n\mathbb{Z})^*$, is $y \in \langle x \rangle$? Give an interactive proof system deciding SubgroupMembership. Is the protocol zero-knowledge?

Proof. See [13, Ch.13.2]. ♯

29 References

[1] A. Aho, J. Hopcroft, J. Ullman: The design and analysis of computer algorithms, second printing, Addison-Wesley Series in Computer Science and Information Processing, 1975.

[2] J. Buchmann: Introduction to cryptography, second edition, Undergraduate Texts in Mathematics, Springer, 2004.

[3] H. Cohen: A course in computational algebraic number theory, Graduate Texts in Mathematics 138, Springer, 1993.

[4] H. Cohen, G. Frey et al.: Handbook of elliptic and hyperelliptic curve cryptography, CRC Press Series on Discrete Mathematics and its Applications, 2006.

[5] G. Hardy, E. Wright: An introduction to the theory of numbers, fifth edition, Oxford University Press, 1979.

[6] H. Heuser: Lehrbuch der Analysis, Teil 1, Teubner, 1980.

[7] J. von zur Gathen, J. Gerhard: Modern computer algebra, second edition, Cambridge University Press, 2003.

[8] N. Koblitz: Algebraic aspects of cryptography, Algorithms and Computation in Mathematics 3, Springer, 1998.

[9] N. Koblitz: A course in number theory and cryptography, second edition, Graduate Texts in Mathematics 114, Springer, 1994.

[10] A. Menezes, P. van Oorschot, S. Vanstone: Handbook of applied cryptography, CRC Press Series on Discrete Mathematics and its Applications, 1997.

[11] C. Papadimitriou: Computational complexity, Addison-Wesley, 1995.

[12] D. Stinson: Cryptography, theory and practice, third edition, CRC Press Series on Discrete Mathematics and its Applications 36, 2006.

[13] D. Stinson: Cryptography, theory and practice, first edition, CRC Press Series on Discrete Mathematics and its Applications 36, 1995.

[14] D. Wätjen: Kryptographie — Grundlagen, Algorithmen, Protokolle, 2. Auflage, Spektrum Verlag, 2008.