B) Symmetric Ciphers

B.a) Fundamentals

B.b) Block Ciphers

B.c) Stream Ciphers

B.a) Fundamentals

B.1 Definition

- A mapping

  Enc: $P \times K \rightarrow C$ for which

  φ_k := Enc($\cdot$,k): $P \rightarrow C$ is bijective for each k$\in K$

  is called an *encryption algorithm.* The sets *P*, *K* and *C* are called

  - *P* : plaintext space
  - *K :* key space
  - *C :* ciphertext space

B.1 (continued)

- The mapping Enc(⋅,⋅) induces a set $\{\varphi_k : P \to C \mid k \in K\}$ of $|K|$ bijections. Its elements are called *encryption transformations.*
- Consequently, there exists a further set of $|K|$ bijections $\{\psi_h : C \to P \mid h \in K\}$ with the property that for each $k \in K$ there exists a unique $h \in K$ so that the composition $\psi_h \circ \varphi_k$ equals the identity mapping on P. That is, $\psi_h(\varphi_k(p)) = p$ for each $p \in P$. These bijections are called *decryption transformations.*
- For any fixed $k \in K$ and any $c \in C$ there exists a unique $p \in P$ with Enc(p,k) = c. We define Dec(c,k):= p and call Dec(⋅,⋅) the *decryption algorithm*. Alternatively, Dec may be denoted by Enc^{-1} .

B.1 (continued)

- The 5-tuple

  $(P,K,C,\{\varphi_{\mathrm{k}} : P \to C \mid \mathrm{k}\in K\}, \{\psi_{\mathrm{h}} : C \to P \mid h\in K\})$

  is called an *encryption scheme* (resp., a *cipher*).

B.3 Remark

- In Definition B.1 more generality can be obtained if φ_k := Enc($\cdot$,k): $P \rightarrow C$ is merely assumed to be injective for each k$\in K$, i.e. bijective onto its image φ_k (P).
- An encryption algorithm Enc($\cdot$,$\cdot$) can alternatively be represented by the set of encryption transformations.
- Some authors denote the sets ($\{\varphi_k : P \rightarrow C \mid k\in K\}$, $\{\psi_h : C \rightarrow P \mid h\in K\}$) an *encryption scheme* (resp., a *cipher*).

B.4 Definition

- An encryption algorithm is called *symmetric* if decryption is computationally easy provided that the encryption key is known. In the notion of encryption and decryption transformations this is equivalent to saying that it is computationally easy to compute h = h(k) from k.
- Note: Otherwise we speak of *asymmetric algorithms* or *public key cryptography* (→ Chapter C).

B.5 Example

- Cesar’s cipher:
  - w *P* = *C* = {A,B,…,Z}
  - w *K* = {0,1,…,25}
  - w shift the plaintext alphabet *P* cyclically by k positions to the left, substitute the plaintext letter by the ciphertext letter at the corresponding position.
- Note: Cesar’s cipher is symmetric. Decrypting merely demands the rotation of the ciphertext alphabet by k positions to the right.

B.6 Definition

- An *adversary* (*attacker, enemy, eavesdropper*) tries to defeat an information security service; e.g. he may try to find a key to decrypt a secret message.
- A *passive adversary* is an adversary who is capable only of reading information from an unsecured channel.
- An *active adversary* may also transmit, alter or delete information on an unsecured channel.

B.7 Typical Goals of a Potential Adversary

- Find the decryption key k
- To given ciphertexts $c_1,c_2,\dots,c_N$ find the corresponding plaintexts $p_1,p_2,\dots,p_N$.
- To given plaintexts $p_1,p_2,\dots,p_N$ find the corresponding ciphertexts $c_1,c_2,\dots,c_N$.

Note: For symmetric ciphers the first goal implies the second and the third. Depending on the concrete situation the second goal may be easier to achieve than the first.

B.8 Attacking Cesar's cipher

- The adversary decrypts given ciphertext $c_1,c_2,\ldots,c_N$ with all 26 admissible keys.
- One key yields meaningful plaintext. This is the searched key. (The other keys give meaningless plaintexts.)

Note: a) Because of its small key space it is very easy to break Cesar's cipher.

B.9 An Improved Variant of Cesar’s Cipher

- $P = C = \{A,B,\ldots,Z\}$
- $K = \{\pi \mid \pi : P \to C$ is bijective$\}$
- Enc(p, π):= π(p)

Note:

a) $| K | = 26! \approx 2^{88}$

b) It is not practically feasible to check key by key.

Question: Does this mean that the improved variant of Cesar’s cipher is secure?

B.10 Attacking the Improved Variant of Cesar’s Cipher

- Unless it is very short the most frequent letter in a ‘typical’ English text is “E”.
- → Substitute the letter that occurs most frequently in the encrypted message by plaintext “E”. This reduces the size of the remaining key space by factor 26 from 26! to 25!
- Continue the attack. Try to substitute further (frequently occurring) letters of the encrypted message by probable plaintext letters …
- If these substitutions were correct the attacker knows a fragment of the plaintext message. It should be possible to guess its complement, which is still unknown.

Details: Blackboard

Exercise: Perform this attack practically

B.11 Generic Design Criteria

The attacks from B.8 and B.10 suggest the following requirements:

a) The key space *K* should be so large that an exhaustive key search (i.e. checking all keys) is not practically feasible (← B.8, attacking Cesar's cipher)

b) The encryption algorithm shall not allow attacks that are essentially faster than exhaustive key search (← B.10, attacking an improved variant of Cesar's cipher)

Note:

It is easy to guarantee Requirement a) but usually it is much more difficult to decide whether b) is fulfilled.

The assessment whether b) is fulfilled may vary in the course of the time (← new attacks)

B.12 Affine Encryption (I)

- Identify {A,B,…,Z} with the set Z_{26}:={0,1,…,25}. More precisely, identify the letter A with 0, the letter B with 1, …, and Z with 25.
- Equip Z_{26} with the addition and multiplication modulo 26. Then Z_{26} is a ring.
- Select an integer $m \geq 1$.
- Definition: GL(m,26) denotes the group of all ($m \times m$)-matrices over Z_{26}
- Remark: $M \in$ GL(m,26) iff (det(M) (mod 26)) $\in Z_{26}^{*}$ iff gcd(det(M),26) = 1

B.12 Affine Encryption (II)

- Substitute each letter of the plaintext by the respective element in Z_{26} and group the plaintext into non-overlapping blocks of m consecutive numbers.
- Encryption of a block **p**:

  Enc(**p**,(A(k$_1$),k$_2$)) := A(k$_1$)**p** + k$_2$ (mod 26), i.e.

  - $P = C = Z_{26}^m$
  - $K = GL(m,26) \times Z_{26}^m$
- Decryption:

  Dec(**c**, (A(k$_1$),k$_2$)) = A(k$_1$)$^{-1}$ (**c** - k$_2$) (mod 26)
- Question: Is the affine cipher secure?

B.13 Attacking the Affine Cipher

- Assumption: The attacker knows (plaintext, ciphertext) pairs $(\mathbf{p}_1,\mathbf{c}_1),\ldots, (\mathbf{p}_{m+1},\mathbf{c}_{m+1})$
- Goal: Find the key $(A(k_1),k_2)$
- Fact: If the column vectors $\mathbf{p}_1-\mathbf{p}_{m+1},\ldots,\mathbf{p}_m-\mathbf{p}_{m+1} \in Z_{26}^m$ form a matrix in GL(m,26) the key is uniquely determined. (Otherwise the attacker needs further (plaintext, ciphertext) pairs.)
- The attack requires the inversion of one matrix and one matrix multiplication in GL(m,26).
- Details: Blackboard

B.14 Types of Attacks (characterization with regard to the attacker's knowledge / abilities)

General assumption: The attacker knows the encryption algorithm.

a) *ciphertext-only attack*: The attacker only knows some ciphertext.

Example: B.8 (attacking Cesar's cipher), B.10 (attacking the improved variant of Cesar's cipher)

b) *known plaintext attack*: The attacker knows some corresponding (plaintext, ciphertext) pairs $(p_1,c_1),\ldots, (p_N,c_N)$.

Example: B.13 (attacking the affine cipher)

B.14 (continued)

c) *chosen plaintext attack*: similar to a known plaintext attack but the attacker is able to select plaintexts $p_1,p_2,\ldots, p_N$.

A chosen-plaintext attack is called *adaptive* if the choice of p_{k+1} depends on $(p_1,c_1),\ldots, (p_k,c_k)$ for $k = 1,2,\ldots, N-1$.

d) *chosen ciphertext attack*: pendant to a chosen plaintext attack where the attacker is able to select the ciphertext

B.15 Remark

a) Ciphertext-only attacks are usually only successful against very weak ciphers, due to inappropriate conditions of use, security flaws in protocols etc.

b) To perform a chosen plaintext attack (resp. a chosen ciphertext attack) the adversary must have access to the encryption device (e.g., a smart card or a server) at least for a period of time and the ability / permission to use it.

B.16 Unconditional Security

An encryption algorithm Enc: $P \times K \rightarrow C$ is said to be *unconditionally secure* (resp., *perfectly secure*) if the knowledge of the ciphertext gives an adversary with unlimited computational power no additional information on the plaintext.

Note: This means

Prob(plaintext=p | ciphertext=c) = Prob(plaintext=p)

for all (p,c) $\in P \times C$

B.17 Remark

Unconditional security is an very strong requirement. All the widespread algorithms are not unconditionally secure (cf. B.23)

B.18 Computational Security

An encryption algorithm Enc: $P \times K \rightarrow C$ is said to be *computationally secure* (resp., *practically secure*) if an attacker is not even able to perform the best currently known attack with non-negligible success probability since the perceived level of computation required to defeat it exceeds, by a comfortable security margin, the computational resources of the hypothesized adversary.

Note: The statement may be restricted (e.g.: "… is computationally secure against known plaintext attacks").

B.19 Further Notions of Security

- complexity-based security
- provable security

  (cf. the “Handbook of Applied Cryptography”, for instance)

B.20 Remark

a) The characterization of computational security is not precise in a mathematical sense.

b) Problem / Difficulty: Designers and evaluators of encryption algorithms may overlook effective attacks.

c) The assessment whether an encryption algorithm is viewed to be computationally secure usually changes in the course of the time.

B.20 (continued)

d) Ideally, new algorithms should be evaluated by a large number of experts. At least *all known types of attacks* should be considered.

e) Sometimes the resistance of an encryption algorithm against specific types of attacks can be proven in a strict sense.

B.21 Composition of Ciphers

- Assume that Enc_1: $P \times K \rightarrow C$ and Enc_2: $C \times K^* \rightarrow C^*$ are encryption algorithms.
- The composition $\mathrm{Enc}_2 \circ \mathrm{Enc}_1$: $P \times (\mathrm{K} \times K^*) \rightarrow C^*$ is also an encryption algorithm.

Notation: $\mathrm{Enc}_2 \circ \mathrm{Enc}_1 (p,(k,k^*)) := \mathrm{Enc}_2 (\mathrm{Enc}_1 (p,k)),k^*)$

Remark: In general, the composition $\mathrm{Enc}_2 \circ \mathrm{Enc}_1$ is stronger than Enc_1 and Enc_2, respectively.

Exercise: Show that the strength of the composition of two Cesar’s ciphers, resp. of two improved Cesar’s ciphers, does not exceed the strength of one cipher of the respective type.

B.22 Remark

When composing encryption algorithms one usually performs three instead of two consecutive encryptions. The reason will be explained in Section B.b.

B.23 One-time pad

plaintext $p_1,p_2,\ldots,p_N \in P = \{0,1\}$

key bits $k_1,k_2,\ldots,k_N \in \{0,1\}$, i.e. $K = \{0,1\}^N$

Assumption / Mathematical model: The key bits $k_1,k_2,\ldots$ are viewed as values that are taken on by independent random variables that are uniformly distributed on {0,1}. (The key bits might be generated by tossing a fair coin, for instance.)

B.23 (continued)

Encryption: $c_j = p_j \oplus k_j$ (= $p_j + k_j$ (mod 2)) for j=1,2,…,N

Decryption: $p_j = c_j \oplus k_j$ for j=1,2,…,N

Security: The knowledge of the ciphertext $(c_1,c_2,\ldots,c_N) \in \{0,1\}^N$ does not give any additional information on the corresponding plaintext: In fact, all keys are equally likely and to each plaintext $p'_1,p'_2,\ldots,p'_N$ there exists exactly one key $k' \in \{0,1\}^N$ with $\mathrm{Dec}(c_1,c_2,\ldots,c_N , k') = p'_1 , p'_2,\ldots,p'_N$. The one-time pad cipher is *unconditionally secure* against decryption attacks.

B.23 (continued)

Disadvantages / Problems:

- The key is as long as the plaintext.
- The key must not be used twice (→ Exercises).
- Consistency “demands” unconditional secure key exchange (e.g. by a trustworthy courier). At least for open networks this is very inconvenient.

B.23 (continued)

Note:

- The one-time pad does not ensure data integrity against active adversaries. Altering particular ciphertext bits results in wrong plaintext bits at these positions after decryption.
- If the attacker knows the structure of the plaintext (e.g., a bank transfer) he may alter particular bits hoping that these changes give a meaningful plaintext (e.g. another valid target account number).

Example: Exercise