**IS ZA SecKeyCrypto**

**Block ciphers:**

* Data Encryption Standard (DES)
* Advanced Encyrption Standard (AES)
* Basic block cipher modes: ECB and CBC

**Stream Ciphers:**

* OFB and CTR

Block ciphers:

* Plaintext wird in beliebig langen Blöcken (n bits) aufgeteilt
* Die Plaintext-Blöcke werden mit dem Block cipher zu Ciphertext-Blöcke konvertiert

Herkömmliche Block Sizes:

* 64 oder 128 bits
* Heute 128 bit

Block Padding:

* Plaintexte sind normalerweise nicht Vielfaches von Block Grössen
* Am Ende des Plaintextes entsprechend füllen => padding

DES

* Well built
* 64 bit = >reduced to 56 bit
* Brute force best known attack

3DES

* 112 bit not 168 bit
* Meet in the Middle Attack:
  + In zwei teilen
    - Zwei DES Operationen
      * 2^112 verschiedene Keys ausprobieren
    - Eine DES Operation
      * 2^56 verschiedene Keys ausprobieren
  + 2^112 + 2^56 fast gleich 2^112

AES

* Secret key (symmetric) block cipher
* Block size = 128bits
* Flexible key size => 128, 192, 256 bits
* Freely available

ECB (Electronic code book mode)

* Same block of ciphertext always decrypts to the same block of plaintext => Schwachstelle
* Angreifer kann die redundanten Blöcke entschlüsseln und dadurch einen Teil des Textes entschlüsseln, weil ja die Blöcke vlt mehrmals vorkommen.
* Every block is treated independently from all others.
  + Leicht manipulierbar

CBC (Cipher Block Chaining Mode)

* Sender: Each plaintext block is XOR-ed with the previous ciphertext block before encryption
  + Identische Plaintext – Blöcke ergeben nicht mehr den gleichen Ciphertext
* Receiver: each block coming out of the decryption must first be XOR-ed with the previously received ciphertext block in order to recover the plaintext.
* Initialisierung => Initialisierungs-Vektor
  + Openly transmited over the insecure channel at the beginning of the session
  + Against replay attacks => IV value should be used only once and never be used again. => use random value or increasing counter
* Same attack as in ECB is possible

Stream Ciphers

* based on a keystream generator => produces pseudo-random keystream initialized by a secret key
* Unterschied **one-time-pad ⬄ stream cipher**
  + Keystream of a one-time pad is truly random by definition
  + One-time pad does not depend on any key
  + Stream cipher depends on a secret key
* A keystream should never be reused, same problem with the one-time pad

Stream Ciphers versus Block Ciphers

* Stream ciphers:
  + Bit-level architecture
  + Very high throughputs can be achieved.
  + Single bit errors in the ciphertext affect only a single plaintext bit and do not propagate
  + Easily manipulated
  + Suitable for real-time communication
    - One does not have to wait for a full block of data
    - Faster than block ciphers
* Block cipher :
  + Word-level architecture
  + Single bit errors propagate
    - Affect consecutive plaintext blocks in CBC mode.

Output Feedback Mode (OFB)

* Keystream generator producing a pseudo-random keystream one block at a time.
* By XOR-ing the keystream with the plaintext, the block sipher works as a stream cipher
* Individual block cipher boxes still receive the shared secret key as an input
* Generated keystream is based on and is unambiguously determined by the shared secret key
* Advantage: use of a well-established block cipher that has demonstrated to produce ciphertext that is statistically independent of the plaintext to produce a highly random keystream without having to rely on another keystream generator

Stream cipher can be implemented with block ciphers:

* OFB
* Counter Mode (CTR)