



Chapter 2

Basics of Cryptography

- ❑ Overview Cryptographic Algorithms
- ❑ Attacking Cryptography
- ❑ Properties of Encryption Algorithms
- ❑ Classification of Encryption Algorithms

Cryptographic Algorithms: Overview



- ❑ During this course two main applications of cryptographic algorithms are of principal interest:
 - ❑ **Encryption** of data: transforms plaintext data into ciphertext in order to conceal its' meaning
 - ❑ **Signing** of data: computes a *check value* or *digital signature* to a given plain- or ciphertext, that can be verified by some or all entities being able to access the signed data
- ❑ Some cryptographic algorithms can be used for both purposes, some are only secure and / or efficient for one of them.
- ❑ Principal categories of cryptographic algorithms:
 - ❑ **Symmetric cryptography** using 1 key for en-/decryption or signing/checking
 - ❑ **Asymmetric cryptography** using 2 different keys for en-/decryption or signing/checking
 - ❑ **Cryptographic hash functions** using 0 keys (the “key” is not a separate input but “appended” to or “mixed” with the data).

Attacking Cryptography (1): Cryptanalysis



- ❑ *Cryptanalysis* is the process of attempting to discover the plaintext and / or the key
- ❑ Types of cryptanalysis:
 - ❑ *Ciphertext only*: specific patterns of the plaintext may remain in the ciphertext (frequencies of letters, digraphs, etc.)
 - ❑ *Known ciphertext / plaintext pairs*
 - ❑ *Chosen plaintext or chosen ciphertext*
 - ❑ Newer developments: *differential cryptanalysis, linear cryptanalysis*
- ❑ Cryptanalysis of public key cryptography:
 - ❑ The fact that one key is publicly exposed may be exploited
 - ❑ Public key cryptanalysis is more aimed at breaking the cryptosystem itself and is closer to pure mathematical research than to classical cryptanalysis
 - ❑ Important directions:
 - Computation of discrete logarithms
 - Factorization of large integers

Attacking Cryptography (2): Brute Force Attack



- ❑ The *brute force attack* tries every possible key until it finds an intelligible plaintext:
 - ❑ Every cryptographic algorithm can in theory be attacked by brute force
 - ❑ On average, half of all possible keys will have to be tried

Average Time Required for Exhaustive Key Search

Key Size [bit]	Number of keys	Time required at 1 encryption / μs	Time required at 10^6 encryption / μs
32	$2^{32} = 4.3 * 10^9$	$2^{31} \mu\text{s} = 35.8 \text{ minutes}$	2.15 milliseconds
56	$2^{56} = 7.2 * 10^{16}$	$2^{55} \mu\text{s} = 1142 \text{ years}$	10.01 hours
128	$2^{128} = 3.4 * 10^{38}$	$2^{127} \mu\text{s} = 5.4 * 10^{24} \text{ years}$	$5.4 * 10^{18} \text{ years}$

Attacking Cryptography (3): How large is large?



Reference Numbers Comparing Relative Magnitudes

Reference	Magnitude
Seconds in a year	$\approx 3 * 10^7$
Seconds since creation of solar system	$\approx 2 * 10^{17}$
Clock cycles per year (1 GHz computer)	$\approx 3.2 * 10^{16}$
Binary strings of length 64	$2^{64} \approx 1.8 * 10^{19}$
Binary strings of length 128	$2^{128} \approx 3.4 * 10^{38}$
Binary strings of length 256	$2^{256} \approx 1.2 * 10^{77}$
Number of 75-digit prime numbers	$\approx 5.2 * 10^{72}$
Electrons in the universe	$\approx 8.37 * 10^{77}$

Important Properties of Encryption Algorithms



- ❑ Consider, a sender is encrypting plaintext messages P_1, P_2, \dots to ciphertext messages C_1, C_2, \dots
Then the following properties of the encryption algorithm are of special interest:
 - ❑ *Error propagation* characterizes the effects of bit-errors during transmission of ciphertext to reconstructed plaintext P_1', P_2', \dots
 - ❑ Depending on the encryption algorithm there may be one or more erroneous bits in the reconstructed plaintext per erroneous ciphertext bit
 - ❑ *Synchronization* characterizes the effects of lost ciphertext data units to the reconstructed plaintext
 - ❑ Some encryption algorithms can not recover from lost ciphertext and need therefore explicit re-synchronization in case of lost messages
 - ❑ Other algorithms do automatically re-synchronize after 0 to n (n depending on the algorithm) ciphertext bits

Classification of Encryption Algorithms: Three Dimensions

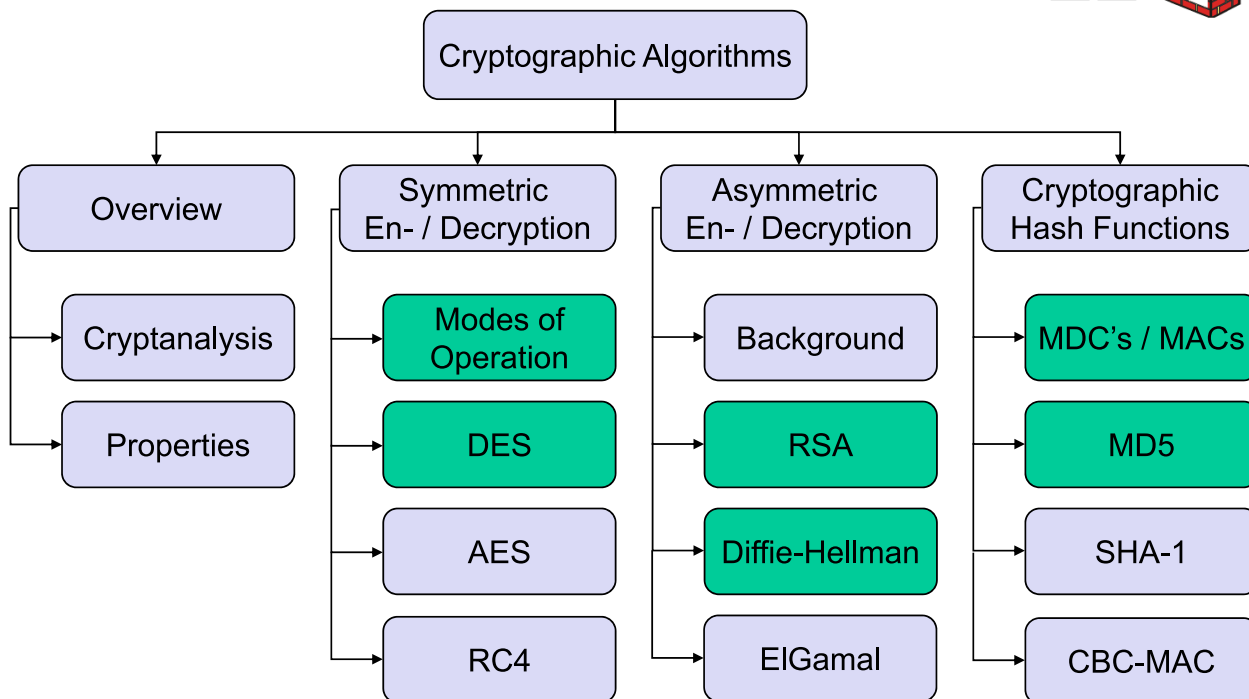


- ❑ The type of operations used for transforming plaintext to ciphertext:
 - ❑ **Substitution**, which maps each element in the plaintext (bit, letter, group of bits or letters) into another element
 - ❑ **Transposition**, which re-arranges elements in the plaintext
- ❑ The number of keys used:
 - ❑ **Symmetric ciphers**, which use the same key for en- / decryption
 - ❑ **Asymmetric ciphers**, which use different keys for en- / decryption
- ❑ The way in which the plaintext is processed:
 - ❑ **Stream ciphers** work on bit streams and encrypt one bit after another:
 - Many stream ciphers are based on the idea of linear feedback shift registers, and there have been detected vulnerabilities of a lot of algorithms of this class, as there exists a profound mathematical theory on this subject.
 - Most stream ciphers do not propagate errors but are sensible to loss of synchronization.
 - ❑ **Block ciphers** work on blocks of width b with b depending on the specific algorithm.

Key Management



- ❑ Key generation
 - ❑ Must use (pseudo) random number generators
 - ❑ Key generation for asymmetric encryption depends on the factorization of large integer numbers
- ❑ Key distribution
 - ❑ Simplest case: personal contact
 - ❑ Encrypted channel for key distribution -> Key hierarchies
- ❑ Key storage
 - ❑ Optimum case: in the brain of the user
 - ❑ Alternatively, in secured crypto modules
- ❑ Key recovery
 - ❑ Simple case: using a saved copy (implicates new security issues)
 - ❑ Alternatively, fragment the key into several sub-keys
- ❑ Key invalidation
 - ❑ Especially required for asymmetric mechanisms
- ❑ Key deletion
 - ❑ Disablement of old encrypted texts



Summary (what do I need to know)



- ❑ Categories of cryptographic algorithms
 - ❑ Symmetric encryption
 - ❑ Asymmetric encryption
 - ❑ Cryptographic hash functions
- ❑ Application of encryption techniques
 - ❑ Encryption
 - ❑ Signing
- ❑ Classification of encryption algorithms
 - ❑ Symmetric vs. asymmetric
 - ❑ Stream vs. block ciphers