Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Network Security

Chapter 2
Basics of Cryptography

- Overview of Cryptographic Algorithms
- Attacking Cryptographic Algorithms
- Historical Approaches
- Foundations of Modern Cryptography

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	$2^{32} = 4.3 * 10^9$	$2^{31}\ \mu s$ = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 * 10^{16}$	$2^{55}\ \mu s$ = 1142 years	10.01 hours
128	$2^{128} = 3.4 * 10^{38}$	$2^{127}\ \mu s = 5.4 * 10^{24}$ years	$5.4 * 10^{18}$ years

- 1 encryption / μs: 100 Clock cycles of a 100 MHz processor
- 10^6 encryptions / μs: Clock cycles using 500 parallel 2GHz processors

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 (3 GHz computer)	$\approx 1 * 10^{17}$
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	$\approx 8.37 * 10^{77}$

Classification of modern encryption algorithms

- 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
  - *Block ciphers* work on blocks of width *b* with *b* depending on the specific algorithm.

Basic Kryptographic Principles

- Substitution
  - Individual characters are exchanged by other characters

  Types of substitution
  - simple substitution substitution: operates on single letters
  - polygraphic substitution: operates on larger groups of letters

  - monoalphabetic substitution: uses fixed substitution over the entire message
  - polyalphabetic substitution: uses different substitutions at different sections of a message

- Transposition
  - The position of individual characters changes (Permutation)

Transposition: scytale

- Known as early as 7th century BC
- Principle:
  - Wrap parchment strip over a wooden rod of a fixed diameter and write letters along the rod.
  - Unwrap a strip and “transmit”
  - To decrypt, wrap a received over a wooden rod of the same diameter and read off the text.

- Example:

```
troops
headii
nthewe      ⇨     thnsm predd opoah nrlod eeeis iedus
stneed
moresu
pplies
```

- Weakness:
  - Easy to break by finding a suitable matrix transposition.

Monoalphabetic substitution: Atbash

Jeremiah 25:25

And all the kings of the north, far and near, one with another, and all the kingdoms of the world, which are upon the face of the earth: and the king of Sheshach shall drink after them.

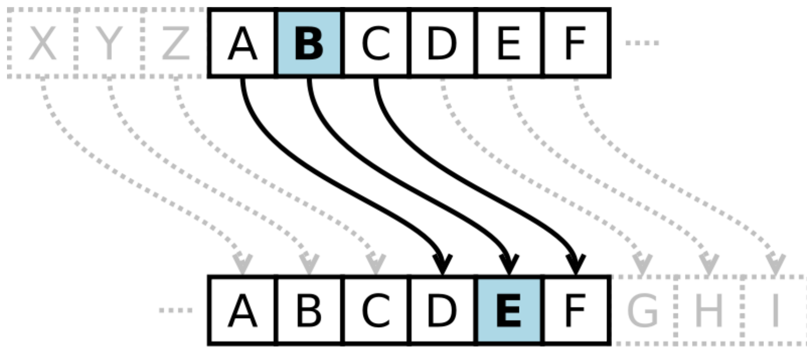
Atbash code: reversed Hebrew alphabet.

A Aleph א	B Beth ב	G Gimel ג	D Daleth ד	H He ה	WVFY Waw ו	Z Zajin ז	H Chet ח	T Tet ט	IJ Jod י	K Kaph כ ך	L Lamed ל	M Mem מ ם	N Nun נ ן	X Samech ס	O Ajin ע	P Pe פ ף	Z Sade צ ץ	Q Koph ק	R Resch ר	S Sin ש	T Taw ת
T Taw ת	S Sin ש	R Resch ר	Q Koph ק	Z Sade צ ץ	P Pe פ ף	O Ajin ע	X Samech ס	N Nun נ ן	M Mem מ ם	L Lamed ל	K Kaph כ ך	IJ Jod י	T Tet ט	H Chet ח	Z Zajin ז	WVFY Waw ו	H He ה	D Daleth ד	G Gimel ג	B Beth ב	A Aleph א

Sheshach ⇨ ש ש ך כ ⇨ ב ב ל ⇨ Babel

Monoalphabetic substitution: Caesar cipher

- Ceasar code: left shift of alphabet by 3 positions.



- Example (letter of Cicero to Caesar):

  `MDEHV RSNQNRQNV PHDH XHVXNPRQNZP`

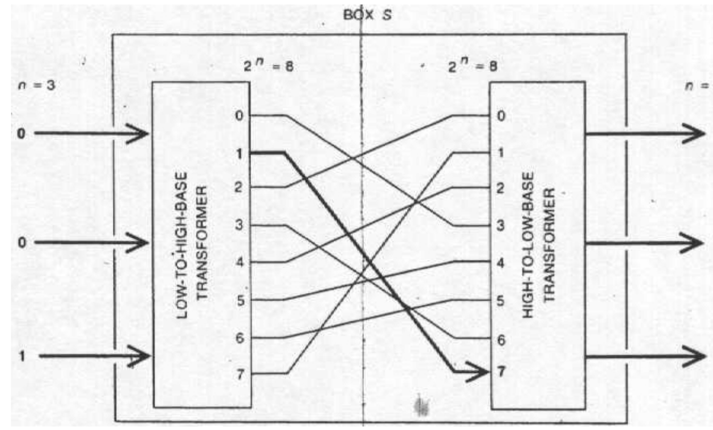
  `HABES OPINIONIS MEAE TESTIMONIUM`

- Weakness: a limited number of possible substitutions. Easy to break by brute force!

Modern cryptography: S and P-boxes

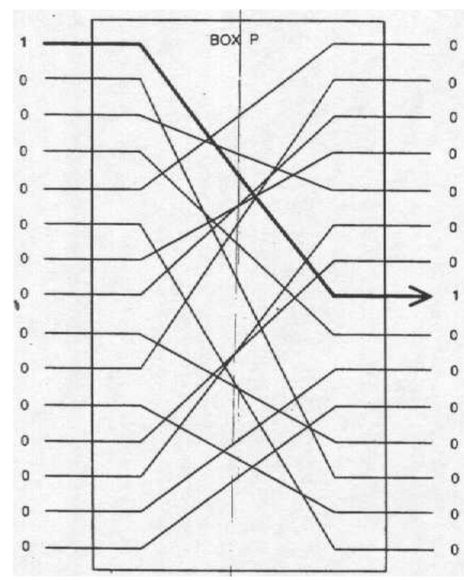
S-box:

- Block-wise **substitution** of binary digits.
- Resistant to attacks for sufficiently large block size; e.g. for n=128 it provides 2^{128} possible mappings.



P-box:

- Block-wise **permutation** of binary digits.
- Realizes a simple **transposition** cipher with maximal entropy.
- Problem: straightforward attacks exist.

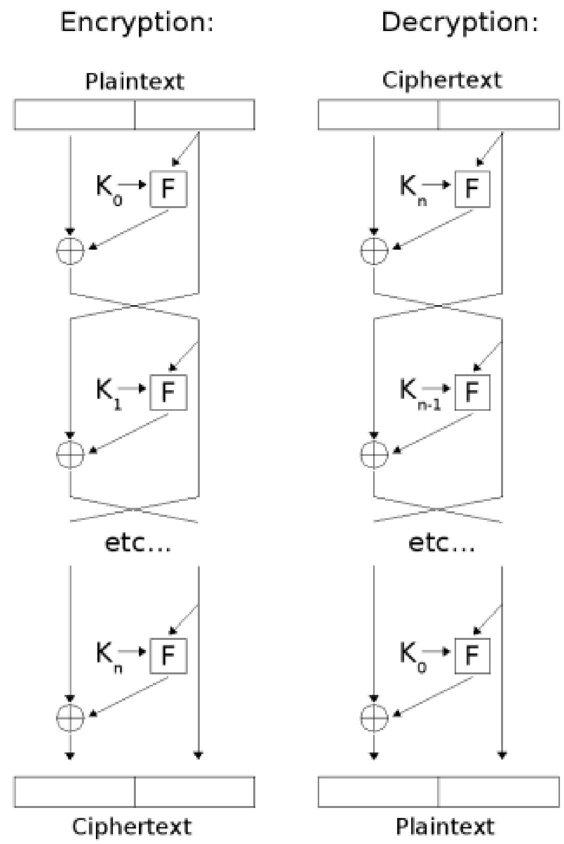


Feistel network: a product cipher of S and P-boxes

- A revival of the idea of a product cipher.
- Multiple rounds provide a cryptographically strong polyalphabetic substitution.
- Combination of substitution with transposition provides protection against specific attacks (frequency analysis).
- Follows the theoretical principles outlined by C. Shannon in 1949: combines “confusion” with “diffusion” to attain maximal entropy of a cipher text.
  - Confusion: cipher text statistics depend in a very complex way on plaintext statistics (approach: substitution in different rounds)
  - Diffusion: each digit in plaintext and in key influence many digits of cipher text (approach: many rounds with transposition)

A practical Feistel cipher

- A multiple-round scheme with separate keys per round.
- Invertible via a reverse order of rounds.
- 3 rounds suffice to achieve a pseudorandom permutation.
- 4 rounds suffice to achieve a strong pseudorandom permutation (i.e. it remains pseudorandom to an attacker with an oracle access to its inverse permutation).
- A foundation for a large number of modern symmetric ciphers: DES, Lucifer, Blowfish, RC5, Twofish, etc.



Feistel Cipher

Important properties of encryption algorithms

Consider, a sender is encrypting plaintext messages P_1, P_2, ... to ciphertext messages C_1, C_2, ...

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', ...
  - 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

Cryptographic algorithms: outline