M183 Applikationssicherheit Implementieren
13

By Jürg Nietlispach

Recap # 12

?

Recap # 12

Data Access

- Databases
  - Types
  - How to store, read & modify Data (CRUD)
  - Attacks

- Data-Ressources (HTTP)
  - How to store, read & modify Data
  - Attacks

Data Access & Manipulation – what's next?

Data Integrity …

*“**Data integrity** is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data”*

Integrity Domains

- **Physical** (Storing & Accessing Physical Data)
    - Altered Traffic (Physical Layer, Network Layer)
    - Material fatigue (Hardware)
    - Corrosion (Hardware)
    - Electromechanical faults (Heisenbugs :-)
    - Environmental Hazards
    - …
- **Logical** (Concerns: Correctness & Rationality)
    - Referential Integrity (Application Layer)
    - Entity Integrity (Application Layer)
    - Programm Assertions (Application Layer)
    - …

How to achieve Integrity?

Physical Layer ?

Logical Layer ?

How to achieve Integrity?

Physical Layer

- Checksums
- Hash-Functions (Message verification)
- Error Correcting (Memory & Data Transport)
- **Encryption** (Prevent unwanted Data Alteration and Information gathering by third parties)
- …

Logical Layer

- Entity Integrity (Primary Keys)
- Referential Integrity (Foreign-Keys)
- Check Constraints
- Application Rules / Business Logic
- …

Encryption Domains

Data «in transit» - data being transferred via networks

Issues: Eavesdropping, Data Alteration (Planned & Unplanned) …

Data «at rest» - Information stored on computers and storage devices

Issues: cryptography, brute-force, stolen cipher exts, attacks on encryption keys,
data corruption, data destruction …

Data Encryption - Overview

- Block Ciphers vs Stream Ciphers
- Substitution Ciphers vs Transposition Ciphers
- Monoalphabetic Substitution
- Polyalphabetic Substitution
- Symmetric Key Systems
- Public Key Systems

Block Cipers vs Stream Ciphers

An important distinction in (symmetric) cryptographic algorithms is between **stream** and **block ciphers**.

Stream ciphers convert one symbol of plaintext directly into a symbol of ciphertext.

Block ciphers encrypt a group of plaintext symbols as one **block**. Simple substitution is an example of a **stream cipher**

Block Cipers vs Stream Ciphers 2

Stream Ciphers

+ Speed of transformation: algorithms are linear in time and constant in space

+ Low error propogation: an error in encrypting one symbol likely will not affect subsequent symbols.

- Low diffusion: all information of a plaintext symbol is contained in a single ciphertext symbol

- No Immunity to tampering: easy to insert symbols without detection

Block Ciphers

+ High diffusion: information from one plaintext symbol is diffused into several ciphertext symbols

+ Immunity to tampering: difficult to insert symbols without detection

- Slowness of encryption: an entire block must be accumulated before encryption / decryption can begin

- Error Propagation: An error in one symbol may corrupt the entire block

Transposition vs Substitution Ciphers

In a **transposition cipher**, the units of the plaintext are rearranged in a different and usually quite complex order, but the units themselves are left unchanged.

By contrast, in a **substitution cipher**, the units of the plaintext are retained in the same sequence in the **ciphertext**, but the units themselves are altered.

Transposition Cipher – Rail Fence

Idea: In the rail fence cipher, the plaintext is written downwards on successive "rails" of an imaginary fence, then moving up when we get to the bottom

Example:
Plaintext -> Ciphertext

```
W . . . E . . . C . . . R . . . L . . . T . . . E
. E . R . D . S . O . E . E . F . E . A . O . C .
. . A . . . I . . . V . . . D . . . E . . . N . .
```

```
WECRL TEERD SOEEF EAOCA IVDEN
```

Transposition Cipher – Route Cipher

Idea: the plaintext is first written out in a grid of given dimensions, then read off in a pattern given in the key

Example:
Plaintext -> Ciphertext.
Key: "spiral inwards, clockwise, starting from the top right"

```
W R I O R F E O E
E E S V E L A N J
A D C E D E T C X
```

```
EJXCTEDECDAEWRIORFEONALEVSE
```

Encryption - Monoalphabetic Substitution

Idea: Every Letter in an Alphabet gets exactly one substitution letter!

Variants

- Cesar-Cipher: Shift (= Key) the Alphabet by n Elements
- Scramble the Alphabet: A -> B, B -> F, C -> Y, etc. (but 1:1)
- Use artificial Characters …

Example

Cesar Cipher (Shift = Key: 13, «Key»: NOP…)

Plaintext:

«Hello World»

Ciphertext:

«Uryyb Jbeyq»

Decryption - Monoalphabetic Substitution

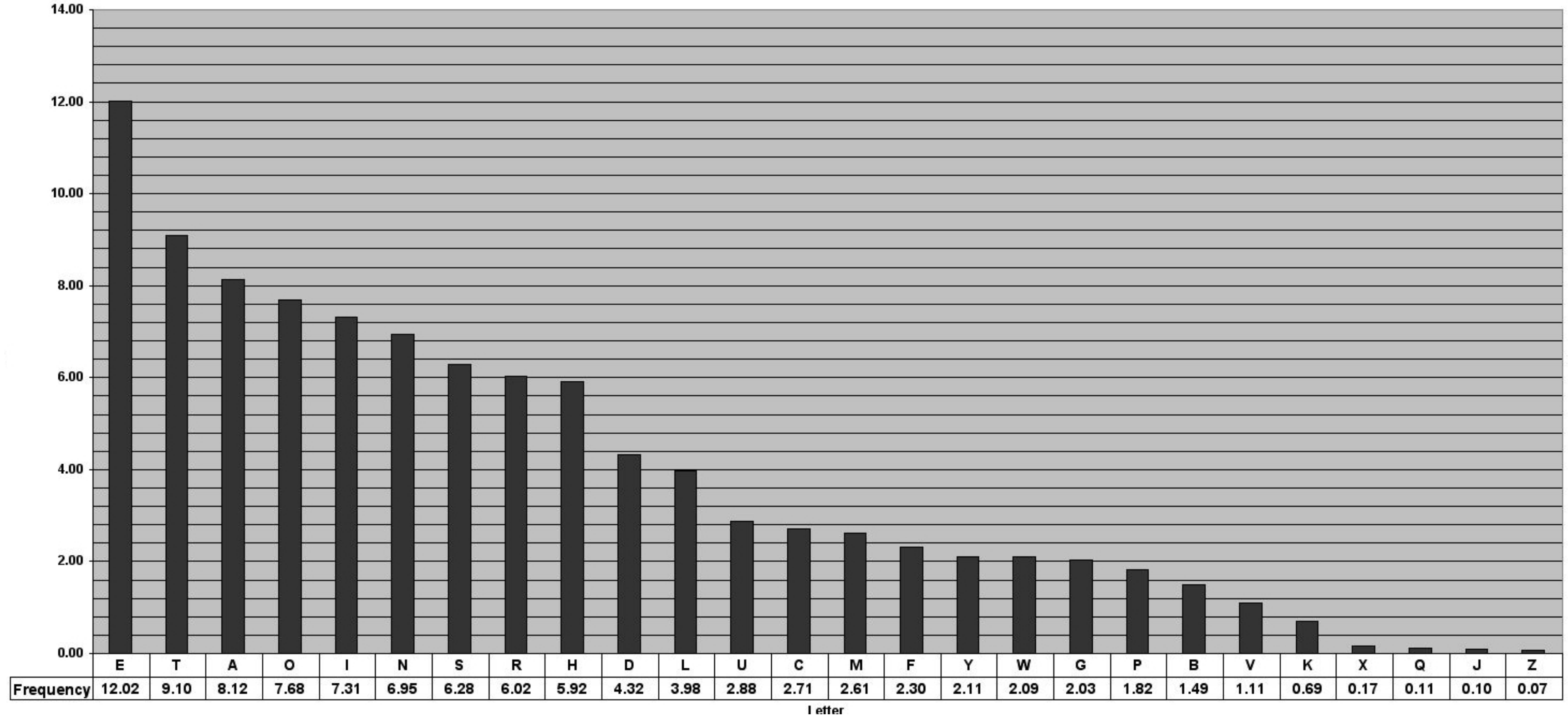
Example
Cesar Cipher (Shift = Key = 13 **backwards** using the **same** «Key»: NOP…)

Ciphertext:
«Uryyb Jbeyq»

Plaintext:
«Hello World»

Cryptoanalysis - Monoalphabetic Substitution - 1

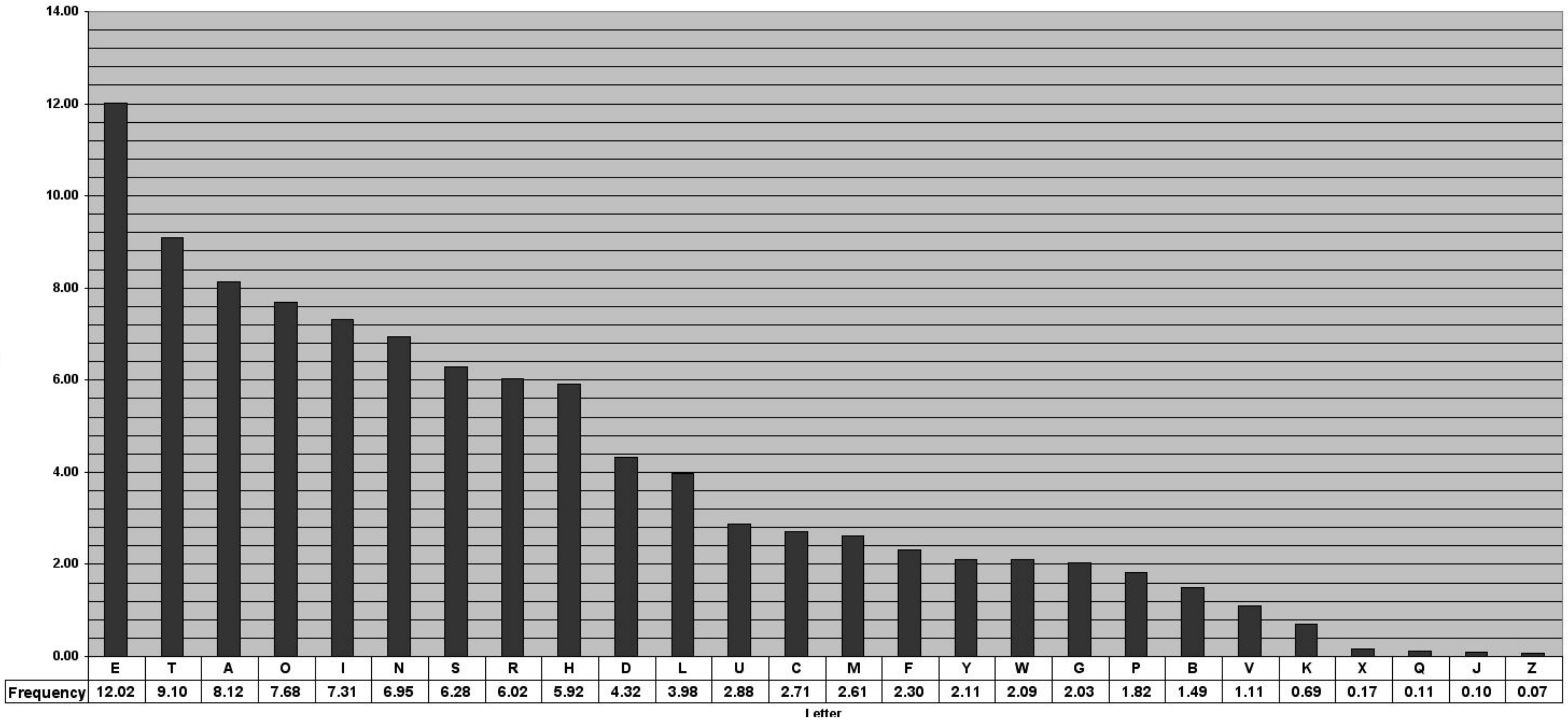
Consider Character Frequencies in English / German Texts



What happens to these Frequencies when doing monoalphabetic substitutions?

Cryptoanalysis - Monoalphabetic Substitution - 2

Only the Letter – Label changes – not the Frequencies!



Instead of E, with a Shift of 13, R will be the most frequent character!

Lab – Monoalphabetic Substitution

1. Monoalphabetic Encryption Engine (Parameter: Shift)
2. Monoalphabetic Cryptoanalysis Engine (Frequency Analysis with Tables / BarChart)
3. Monoalphabetic Decryption Engine (Parameter: Back-Shift)

Encryption - Polyalphabetic Substitution

Idea

One single Character of an alphabet can be mapped to different Characters of the same alphabet.

Example

Vigenère Cipher: using a Key K for encryption of a plaintext

Plaintext: A T T A C K A T D A W N
Key: L e m o n L e m o n L e
Ciphertext: L X F O P V E F R N H R

Letter A is Mapped to L, O, E and N

Decryption - Polyalphabetic Substitution

Example

Vigenère Cipher: using a Key K for decryption of a ciphertext

Ciphertext: L X F O P V E F R N H R
Key: L e m o n L e m o n L e
Plaintext: A T T A C K A T D A W N

Cryptoanalysis - Polyalphabetic Substitution

Idea

The Fact that there are certain mappings of letters or words, that stay the same (a.k.a Repetitions)

Example 1

Vigenère Cipher: using a Key K for encryption of a plaintext

Plaintext: A T T A C K A T D A W N
Key: L e m o n L e m o n L e
Ciphertext: L X F O P V E F R N H R

Cryptoanalysis - Polyalphabetic Substitution

Example 2

Vigenère Cipher: using a Key K for encryption of a plaintext

```
Key:          ABCDABCDABCDABCDABCDABCDABCD
Plaintext:    CRYPTOISSHORTFORCRYPTOGRAPHY
Ciphertext:   CSASTPKVSIQUTGQUCSASTPIUAQJB
```

Key-Length may be: 16 or every factor of 16: 8, 4, 2, 1

Cryptoanalysis - Polyalphabetic Substitution

When the Key Length L is known -> display Ciphertext in Columns of L

L1	L2	L3	L4	L5	L6
C	V	J	T	N	A
F	E	N	M	C	D
M	K	B	X	F	S
T	K	L3	H	G	S
O	J	W	H	O	F

Perform Frequency analysis within each Column and calculate the shift to the regular frequency of the language.
L1 corresponds to the letter of the shifted position.

Lab – Polyalphabetic Substitution

1. Polyalphabetic Encryption Engine (Parameter: Key as Character String)
2. Polyalphabetic Cryptoanalysis Engine (Frequency Analysis with Tables / BarChart)
3. Polyalphabetic Decryption Engine (Parameter: Key as Character String)
4. Perform Frequency Analysis per Key Column