Kryptologie

Programmierkurs in C

Prof. Dr. Robert Lorenz

Lehrprofessur für Informatik
Universität Augsburg

Wintersemester 2017/2018

Kryptographie

Die **Kryptographie** (auch: Kryptografie; von griechisch: „schreiben“) ist die Wissenschaft von sicherer (allgemein geheimer) Kommunikation. Diese Sicherheit bedingt, dass die berechtigten Teilnehmer in der Lage sind, eine Nachricht mit Hilfe eines Schlüssels in einen Geheimtext zu transferieren und zurück. Obwohl der Geheimtext für jemand ohne den geheimen Schlüssel unlesbar und unfälschbar ist, kann der berechtigte Empfänger entweder das Chiffrat entschlüsseln, um die den verborgenen Klartext wieder zu erhalten, oder verifizieren, dass die Nachricht aller Wahrscheinlichkeit nach von jemand geschickt wurde, der den richtigen Schlüssel besaß.

Kryptoanalyse

Die **Kryptoanalyse** (in neueren Publikationen auch: Kryptanalyse) bezeichnet im ursprünglichen Sinne das Studium von Methoden und Techniken, um Informationen aus verschlüsselten Texten zu gewinnen. Diese Informationen können sowohl der verwendete Schlüssel als auch der Originaltext sein. Heutzutage bezeichnet der Begriff Kryptoanalyse allgemeiner die Analyse von kryptographischen Verfahren (nicht nur zur Verschlüsselung) mit dem Ziel, diese entweder zu „brechen“, d.h. ihre Schutzfunktion aufzuheben bzw. zu umgehen, oder ihre Sicherheit nachzuweisen

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Geschichtliche Entwicklung der Kryptologie

- Hieroglyphen
- Tattoo auf dem kahlgeschorenen Kopf eines Sklaven
- 600 v. Ch. Atbash
- 500 v. Ch. Skytale
- 100 - 40 v. Ch. Cäsar: Cäsar-Chiffre
- 1339 - 1365 Rudolf IV: Alphabetum Kaldeorum
- 1404 - 1472 Leon Battista Alberti: Alberti Chiffre
- 1523 - 1596 Blaise de Vigenère: Vigenère Verschlüsselung
- 1883 Auguste Kerckhoffs: Kerckhoffs' Prinzip
- 1918 Gilbert Vernam: One Time Pad
- 1918 Arthur Scherbius: Enigma
- 1976 IBM: DES
- 1977 Rivest, Shamir, Adleman: RSA-Verschlüsselung
- 1999 IBM: 3DES
- 2000 Daemen, Rijmen: AES
- Heute: Quantenverschlüsselung

Die Cäsar-Chiffre ist ein Spezialfall der monoalphabetischen Verschlüsselung.

Die monoalphabetische k-Verschlüsselung verschiebt einen Klartext p um den Buchstaben k (für $k = 'c'$ Cäsar).

Es gilt also:

$$c = (p + k) \bmod N$$

Die Cäsar-Chiffre ist ein Spezialfall der monoalphabetischen Verschlüsselung.
Die monoalphabetische k-Verschlüsselung verschiebt einen Klartext p um den Buchstaben k (für $k = 'c'$ Cäsar).
Es gilt also:

$$c = (p + k) \bmod N$$

Die Cäsar-Chiffre ist ein Spezialfall der monoalphabetischen Verschlüsselung.
Die monoalphabetische k-Verschlüsselung verschiebt einen Klartext p um den Buchstaben k (für $k = 'c'$ Cäsar).
Es gilt also:

$$c = (p + k) \bmod N$$

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):

c: Code

k: Key

N: Mächtigkeit des Schlüsselraumes

p: Plaintext

Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):

c: Code

k: Key

N: Mächtigkeit des Schlüsselraumes

p: Plaintext

Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):

c: Code

k: Key

N: Mächtigkeit des Schlüsselraumes

p: Plaintext

Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):

c: Code

k: Key

N: Mächtigkeit des Schlüsselraumes

p: Plaintext

Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):

c: Code

k: Key

N: Mächtigkeit des Schlüsselraumes

p: Plaintext

Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Cäsar-Chiffre

Wir einigen uns auf folgende Konventionen (KONVENTION I):
c: Code
k: Key
N: Mächtigkeit des Schlüsselraumes
p: Plaintext
Da mir »monoalphabetische Verschlüsselung« zu lang ist, ist im Folgenden (sofern nicht anders erwähnt) mit »Cäsar-Chiffre« eine allgemeine monoalphabetische Verschlüsselung gemeint!

Beispiel

Key: $'R'$

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	...
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	...

Klartext: DASISTEINTEST
Code: URJZJKVZEKVJK
$\Rightarrow$ Key: $'A'$ entspricht keiner Verschlüsselung

Cäsar und C

Wertebereiche:
$[A-Z]$sind ASCII-Zeichen von $65-90$
$[a-z]$sind ASCII-Zeichen von $97-122$

- Sollen/wollen Wertebereiche eingehalten werden?
- $\rightarrow$ Texte bestehen für gewöhnlich auch aus Whitespace, Umlauten und Sonderzeichen

Cäsar und C

Wertebereiche:
$[A-Z]$sind ASCII-Zeichen von $65-90$
$[a-z]$sind ASCII-Zeichen von $97-122$

- Sollen/wollen Wertebereiche eingehalten werden?
- $\rightarrow$ Texte bestehen für gewöhnlich auch aus Whitespace, Umlauten und Sonderzeichen

Cäsar und C

Wertebereiche:
$[A-Z]$sind ASCII-Zeichen von $65-90$
$[a-z]$sind ASCII-Zeichen von $97-122$

- Sollen/wollen Wertebereiche eingehalten werden?
- $\rightarrow$ Texte bestehen für gewöhnlich auch aus Whitespace, Umlauten und Sonderzeichen

KONVENTION II:
Texte (Plaintext, Key, Code) bestehen **ausschließlich** aus den Buchstaben $[A - Z]$

Cäsar und C

Verschiebung / Verschlüsselung:

- Man verschiebt nicht um Key, sondern um $k - 'A'$ (Netto-Wert)
- $N = 26$ (Mächtigkeit des Alphabets [A-Z])
- Beachte: $p + k > 'Z'$ (Überlauf behandeln)

Cäsar und C

Verschiebung / Verschlüsselung:

- Man verschiebt nicht um Key, sondern um $k - 'A'$ (Netto-Wert)
- $N = 26$ (Mächtigkeit des Alphabets [A-Z])
- Beachte: $p + k > 'Z'$ (Überlauf behandeln)

Cäsar und C

Verschiebung / Verschlüsselung:

- Man verschiebt nicht um Key, sondern um $k - 'A'$ (Netto-Wert)
- $N = 26$ (Mächtigkeit des Alphabets [A-Z])
- Beachte: $p + k > 'Z'$ (Überlauf behandeln)

Entschlüsseln

- Umkehrung der Verschlüsselung: $p = c - k$
- Beachte: $p <' A'$

Attacke!

Zwei mögliche Angriffsmethoden auf Cäsar:

- Brute-Force
- Häufigkeitsanalyse

Brute-Force

Der Brute-Force Angriff wendet sukzessive jeden möglichen Schlüssel auf den Code an, bis ein lesbares / vernünftiges Ergebnis zustandekommt (erweiterbar durch Wörterbuch-Vergleich).
Voraussetzung:

- Schlüsselraum muss bekannt sein
- Algorithmus zur Entschlüsselung mittels gefundenen Schlüssel muss bekannt sein

In unserem Beispiel benötigt der Brute-Force Angriff max. 26 (ist also $O(N)$) Durchläufe

Brute-Force in C

- Wähle Schlüssel aus Schlüsselraum
- Entschlüssle Teilcode mit Schlüssel
- Wenn Teilcode Sinn macht, Schlüssel ist vermutlich der gesuchte, sonst: Wähle neuen Schlüssel.
- Entschlüssle Text mit gewähltem Schlüssel

Häufigkeitsanalyse

Voraussetzung:

- Wissen oder Vermutung über die Sprache des Klartexts
- Häufigkeitsverteilung der Buchstaben in dieser Sprache

Häufigkeitsanalyse

Vorgehen 1:

- Zähle das Vorkommen jedes Buchstabens im Code
- Vergleiche den häufigsten Buchstaben mit dem häufigsten Buchstaben der entspr. Sprache
- Die Differenz der beiden Buchstaben im Code und der entspr. Sprache liefert den Schlüssel
- Verschiebe Cäsar um entspr. Wert zurück (=> Algorithmus muss bekannt sein!)

Häufigkeitsanalyse

Vorgehen 2 (v.a. bei Substitutionschiffren geeignet):

- Ordne den häufigsten Buchstaben im Code den häufigsten Buchstaben der Sprache zu
- Ordne den 2.-häufigsten Buchstaben im Code den 2.-häufigsten Buchstaben der Sprache zu
- ...

=> Algorithmus muss also nicht bekannt sein.

Häufigkeitsanalyse in C

(In C nur Variante 1)

- Ein Array *count* der Größe N festlegen und mit 0 initialisieren
- Beim Auftauchen eines Buchstabens p_i muss *count* an der entspr. Stelle hochgezählt werden: $++count[(int)p_i - 65]$ (65 da $(int)'A' = 65$)
- Suche das i mit $count[i]$ maximalem Wert
- Vergleiche i mit $'E'$ und bilde Differenz $\Rightarrow$ Schlüssel k

$$k = (26 + (i - ('E' - 65))) \bmod 26$$

Vigenère ist eine mehrfache Cäsar-Verschlüsselung. Dabei legt man einen Schlüssel k fest. Jeder Buchstabe k_i des Schlüsselwortes ist ein Cäsar-Schlüssel.
Die Länge des Schlüssels k sei l_k.

Vigenère

Beispiel:
Klartext: *DASISTEINTEST* Schlüssel: *AKEY*

D	A	S	I	S	T	E	I	N	T	E	S	T	Klartext
A	K	E	Y	A	K	E	Y	A	K	E	Y	A	Schlüssel
D	K	W	G	S	D	I	G	N	D	I	Q	T	Code
0	10	4	24	0	10	4	24	0	10	4	24	0	Verschiebung

Vigenère in C

Nach der Implementierung von Cäsar in C ist Vigenère analog (l_k - facher Cäsar):

- Hole ersten Buchstaben von Schlüssel k: $k[0]$
- Hole ersten Buchstaben von Klartext p: $p[0]$
- Verschiebe $p[0]$ um $k[0]$ (Cäsar)
- Gehe zu nächsten Buchstaben in Klartext und Schlüssel
  - Falls Schlüsselende erreicht, gehe wieder nach vorne
  - Sonst verschiebe $p[i]$ um $k[i]$ (Cäsar)

Attacke!

Annahme: Wir kennen die Schlüssellänge l_k

Brute-Force

Ausprobieren aller Schlüssel.
=> Voraussetzung: Wissen über die Generierung des Schlüssels.
=> Problem: Worst-Case-Scenario: 26^{l_k} Durchläufe!
Brute-Force kann bei ausreichend Rechenleistung angewendet werden.
Weiteres Problem: Es können mehrere »sinnvolle« Klartexte entstehen (vgl. One-Time-Pad)!
Fazit: Schlechter Angriff.

Brute-Force

Ausprobieren aller Schlüssel.
=> Voraussetzung: Wissen über die Generierung des Schlüssels.
=> Problem: Worst-Case-Scenario: 26^{l_k} Durchläufe!
Brute-Force kann bei ausreichend Rechenleistung angewendet werden.
Weiteres Problem: Es können mehrere »sinnvolle« Klartexte entstehen (vgl. One-Time-Pad)!
Fazit: Schlechter Angriff.

Brute-Force

Ausprobieren aller Schlüssel.
=> Voraussetzung: Wissen über die Generierung des Schlüssels.
=> Problem: Worst-Case-Scenario: 26^{l_k} Durchläufe!
Brute-Force kann bei ausreichend Rechenleistung angewendet werden.
Weiteres Problem: Es können mehrere »sinnvolle« Klartexte entstehen (vgl. One-Time-Pad)!
Fazit: Schlechter Angriff.

Brute-Force

Ausprobieren aller Schlüssel.
=> Voraussetzung: Wissen über die Generierung des Schlüssels.
=> Problem: Worst-Case-Scenario: 26^{l_k} Durchläufe!
Brute-Force kann bei ausreichend Rechenleistung angewendet werden.
Weiteres Problem: Es können mehrere »sinnvolle« Klartexte entstehen (vgl. One-Time-Pad)!
Fazit: Schlechter Angriff.

Brute-Force in C

Klar?

Häufigkeitsanalyse

Die Schlüssellänge l_k ist bekannt => alle Schlüssel im Abstand von l_k Zeichen gehören zur gleichen Cäsar-Chiffre.
Das AKEY-Beispiel ($l_k = 4$):

D	K	W	H	**S**	D	I	H	**N**	S	I	R	**T**	Code

D	**K**	W	H	S	**D**	I	H	N	**S**	I	R	T	Code

D	K	**W**	H	S	D	**I**	H	N	S	**I**	R	T	Code

usw.

Häufigkeitsanalyse

Die Schlüssellänge l_k ist bekannt => alle Schlüssel im Abstand von l_k Zeichen gehören zur gleichen Cäsar-Chiffre.
Das AKEY-Beispiel ($l_k = 4$):

D	K	W	H	**S**	D	I	H	**N**	S	I	R	**T**	Code
D	**K**	W	H	S	**D**	I	H	N	**S**	I	R	T	Code
D	K	**W**	H	S	D	**I**	H	N	S	**I**	R	T	Code

usw.

Häufigkeitsanalyse

Die Schlüssellänge l_k ist bekannt => alle Schlüssel im Abstand von l_k Zeichen gehören zur gleichen Cäsar-Chiffre.
Das AKEY-Beispiel ($l_k = 4$):

D	K	W	H	**S**	D	I	H	**N**	S	I	R	**T**	Code

D	**K**	W	H	S	**D**	I	H	N	**S**	I	R	T	Code

D	K	**W**	H	S	D	**I**	H	N	S	**I**	R	T	Code

usw.

Häufigkeitsanalyse

Die Schlüssellänge l_k ist bekannt => alle Schlüssel im Abstand von l_k Zeichen gehören zur gleichen Cäsar-Chiffre.
Das AKEY-Beispiel ($l_k = 4$):

D	K	W	H	**S**	D	I	H	**N**	S	I	R	**T**	Code

D	**K**	W	H	S	**D**	I	H	N	**S**	I	R	T	Code

D	K	**W**	H	S	D	**I**	H	N	S	**I**	R	T	Code

usw.

Häufigkeitsanalyse

Man fasst also jeweils alle Zeichen im Abstand von l_k zusammen und wendet auf diese eine Häufigkeitsanalyse an

Häufigkeitsanalyse in C

Gegeben ist $l_k$und c => durch $strlen(c)$ ist die Länge von c auch bekannt.

Baue Hilfscode h aus $c_0 c_{l_k} ... c_{i*l_k} = h$

Wende Häufigkeitsanalyse auf h an => 1. Schlüsselbuchstabe k_1

Bauche Hilfscode h aus $c_1 c_{1+l_k} ... c_{1+i*l_k} = h$

Wende Häufigkeitsanalyse auf h an => 2. Schlüsselbuchstabe k_2

...

Bauche Hilfscode h aus $c_{l_k-1} c_{l_k-1+l_k} ... c_{l_k-1+i*l_k} = h$

Wende Häufigkeitsanalyse auf h an => l_k. Schlüsselbuchstabe k_{l_k}

=> Schlüssel $k_1 k_2 ... k_{l_k}$

Häufigkeitsanalyse in C

Gegeben ist $l_k$und c => durch $strlen(c)$ ist die Länge von c auch bekannt.
Baue Hilfscode h aus $c_0 c_{l_k} ... c_{i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 1. Schlüsselbuchstabe k_1
Bauche Hilfscode h aus $c_1 c_{1+l_k} ... c_{1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 2. Schlüsselbuchstabe k_2
...
Bauche Hilfscode h aus $c_{l_k-1} c_{l_k-1+l_k} ... c_{l_k-1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => l_k. Schlüsselbuchstabe k_{l_k}
=> Schlüssel $k_1 k_2 ... k_{l_k}$

Häufigkeitsanalyse in C

Gegeben ist $l_k$und c => durch $strlen(c)$ ist die Länge von c auch bekannt.
Baue Hilfscode h aus $c_0 c_{l_k} ... c_{i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 1. Schlüsselbuchstabe k_1
Bauche Hilfscode h aus $c_1 c_{1+l_k} ... c_{1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 2. Schlüsselbuchstabe k_2
...
Bauche Hilfscode h aus $c_{l_k-1} c_{l_k-1+l_k} ... c_{l_k-1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => l_k. Schlüsselbuchstabe k_{l_k}
=> Schlüssel $k_1 k_2 ... k_{l_k}$

Häufigkeitsanalyse in C

Gegeben ist $l_k$und c => durch $strlen(c)$ ist die Länge von c auch bekannt.
Baue Hilfscode h aus $c_0 c_{l_k} ... c_{i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 1. Schlüsselbuchstabe k_1
Bauche Hilfscode h aus $c_1 c_{1+l_k} ... c_{1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 2. Schlüsselbuchstabe k_2
...
Bauche Hilfscode h aus $c_{l_k-1} c_{l_k-1+l_k} ... c_{l_k-1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => l_k. Schlüsselbuchstabe k_{l_k}
=> Schlüssel $k_1 k_2 ... k_{l_k}$

Häufigkeitsanalyse in C

Gegeben ist $l_k$und c => durch $strlen(c)$ ist die Länge von c auch bekannt.
Baue Hilfscode h aus $c_0 c_{l_k} ... c_{i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 1. Schlüsselbuchstabe k_1
Bauche Hilfscode h aus $c_1 c_{1+l_k} ... c_{1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => 2. Schlüsselbuchstabe k_2
...
Bauche Hilfscode h aus $c_{l_k-1} c_{l_k-1+l_k} ... c_{l_k-1+i*l_k} = h$
Wende Häufigkeitsanalyse auf h an => l_k. Schlüsselbuchstabe k_{l_k}
=> Schlüssel $k_1 k_2 ... k_{l_k}$

Zum Nachdenken und...

Gerade ist man davon ausgegangen, dass l_k bekannt ist:

- Was passiert, wenn l_k nicht bekannt?
- Kann man l_k irgendwie ermitteln?
- Falls ja: Wie?

Kasiski

- Buchstabenfolgen wiederholen sich (Suffixe, Präfixe, Wortstämme, ...).
- Die Wahrscheinlichkeit von wiederholt gleichen Verschlüsselungen ist hoch.
- Vermutung: Abstände zwischen sich wiederholenden Zeichenfolgen sind Vielfaches der Schlüssellänge

Idee von Kasiski

- Finde sich wiederholende Zeichenfolgen im Text
- Notiere jeweils die Abstände zwischen sich wiederholenden Zeichenfolgen
- Sortiere die gefundenen Abstände nach ihrer Häufigkeit
- Wähle die N häufigsten
- Bilde den ggT, welcher dann die vermutete Schlüssellänge ist.