13 Symetric encryption

Peter Rutschmann

05.11.2025

Symetrische Verschlüsselung

Bei der **symmetrischen Verschlüsselung** wird **derselbe Schlüssel** sowohl zum **Verschlüsseln** als auch zum **Entschlüsseln** einer Nachricht verwendet.

> Sender und Empfänger müssen denselben geheimen Schlüssel kennen und sicher austauschen.

Sie konnten bereits Erfahrungen mit den beiden Verfahren **Ceasar-** und **Viegenere-Verschlüsselung** machen.

- Beide Verfahren verwenden einen Schlüssel:
  - Ceasar-Verschlüsselung: Zahl: um wieviele Buchstaben das Alphabet verschoben wird.
  - Vigenere: Key-Wort: wie stark jeder Buchstabe einzeln verschoben wird.
- Bei beiden Verfahren werden die Buchstaben nach einer bestimmten Regeln durch andere Buchstaben ersetzt. Dies nennt man **Substitution**.

  Substitution bedeutet in der Kryptografie, dass Zeichen oder Bitmuster systematisch durch andere ersetzt werden – die Position bleibt gleich, nur der Inhalt ändert sich.

Aufgabe warming up mit Ceasar

```
\NormalTok{Verschlüsselung: Ceasar}
\NormalTok{Alphabet: abcdefghijklmnopqrstuvwxyz}
\NormalTok{Key: 5}
\NormalTok{Verschlüsselter Text: fqxjwfltsfzkijwoflijnsjsgqfzjsxyjnsknsijy}

\NormalTok{Wie lautet der entschlüsselte Text?}
```

Lösung

alseragonaufderjagdeinenblauensteinfindet
abcdefghijklmnopqrstuvwxyz 12345abcdefghijklmnopqrstuvwxyz vwxyz
fqxjwflts… alseragon –> als eragon

Aufgabe warming up mit Vigenere

```
\NormalTok{Verschlüsselung: Vigenere}
\NormalTok{Alphabet: abcdefghijklmnopqrstuvwxyz}
\NormalTok{Key: saphira}
\NormalTok{Verschlüsselter Text:
shcaminacwalrskdxlavrxuckmznwssyithwntpmjswicsmsefvtyivnveguezrv}

\NormalTok{Wie lautet der entschlüsselte Text?}
```

Lösung

ahnternichtdassdieserfundeinesdracheneiesseinlebenveraendernwird

---

Die Caesar- und die Viegenere-Verschlüsselung lassen sich mit etwas Aufwand entschlüsseln, auch wenn man den Schlüssel nicht kennt. Nutzt man einen Computer, so dauert das Knacken des Codes Bruchteile von Sekunden.

Eine sichere Verschlüsselung muss also ein Verfahren anwenden, so dass man den verschlüsselten Inhalt ohne Kenntnisse des Schlüssels nicht innert nützlicher Zeit knacken kann. Das ist eine Herausforderung, deren sich Mathematiker annehmen.

Symmetrische Verschlüsselungsverfahren in der Informatik

Für die Informatik, sind Caesar und Vigenère nicht geeignet. Computer können diese Codes in Sekunden durch Häufigkeitsanalyse oder Brute-Force entschlüsseln.

Mögliche symetrische Verschlüsselungsverfahren der Informatik sind:

Verfahren	Beschreibung	Autor	Sicherheit
3DES (Triple DES)	Dreifache DES-Verschlüsselung	IBM (basiert auf DES)	Nicht mehr empfohlen (langsam, teilweise unsicher)
AES (Advanced Encryption Standard)	Blockcipher, heute globaler Standard	Joan Daemen & Vincent Rijmen (Rijndael)	Sehr sicher (empfohlen)
Blowfish	Blockcipher, 64-bit Blockgröße	Bruce Schneier	Sicher, aber Blockgröße heute problematisch
Camellia	AES-ähnliche Blockcipher	Mitsubishi & NICT Japan	Sehr sicher, weniger verbreitet
ChaCha20	Stromchiffre, modern & schnell	Daniel J. Bernstein	Sehr sicher, bevorzugt in TLS/HTTPS & VPNs
DES (Data Encryption Standard)	Blockcipher, 56-bit Schlüssel	IBM (mit NSA Einfluss)	Unsicher (zu kurzer Schlüssel)
IDEA	Blockcipher, 128-bit Schlüssel	Lai & Massey	Sicher, aber patentiert (lange Zeit)
RC4	Stromchiffre	Ronald Rivest	Unsicher (nicht mehr verwenden)
RC5	Blockcipher mit variabler Block- & Schlüssellänge	Ronald Rivest	Moderate Sicherheit (alt)
RC6	AES-Finalist, Blockcipher	Ronald Rivest, RSA Labs	Sicher, aber weniger verbreitet als AES
Salsa20	Vorgänger von ChaCha20	Daniel J. Bernstein	Sehr sicher, performant
Serpent	Blockcipher, AES-Finalist	Ross Anderson, Eli Biham, Lars Knudsen	Sehr sicher, aber langsamer als AES
Twofish	Blockcipher, AES-Finalist	Bruce Schneier & Team	Sehr sicher, Alternative zu AES

Binäre Darstellung einer Nachricht

Alle digitalen Informationen sind binär. Texte, Bilder, Audio – alles wird im Computer als Folge von Nullen und Einsen gespeichert und verarbeitet.

Digitale Verschlüsselungsverfahren arbeiten immer auf dieser binären Darstellung einer Nachricht – also auf Bits (0 und 1).

> Bit: 0 oder 1 Nibble: 4 Bit: => 0000 bis 1111 Byte: 8 Bit: => 0000’0000 bis 1111’1111 (Das ’ Zeichen hilft beim Lesen)

Ein symetrisches Verschlüsselungsverfahren nimmt diese Bitfolgen als Eingabe (Klartext) und transformiert sie anhand des Schlüssel und seinem Algorythmus zu einer neuen Bitfolge (Geheimtext).

Ein Text muss also als erstes in eine binäre Zeichenfolge umgewandelt werden.

Beispiel:

```
\NormalTok{Klartext:         H       E        L        L        O}
\NormalTok{Binär (UTF8): 01001000 01000101 01001100 01001100 01001111}

\NormalTok{Verschlüsselt: 10111001 01101100 11010100 ...}
```

Binar versus Hexadezimal

Menschen können Hexwerte einfacher erkennen und vergleichen als lange Bitfolgen. Hexwerte sind Zahlen im 16er Format. Speicheradressen, Farbwerte, Zeichencodes und verschlüsselte Daten werden fast immer in Hex dargestellt.

```
\NormalTok{Buchstabe: A}
\NormalTok{UTF8{-}Code: 48 (hex), 72(dez), 01001000 (bin)}
```

Die beiden Zeichen aus dem Hexwert entsprechen jeweils 4 Bit des Bytes. Damit lässt sich eine Bitfolge einfach in einen Hexfolge umrechnen. (und umgekehrt)

```
\NormalTok{  4    8}
\NormalTok{0100\textquotesingle{}1000}
```

Aufgabe Textzeichen als hexadezimaler Code

Welchem hexadezimalen Code entspricht das durchbare Zeichen B? Dies ist in einer Tabelle fix festgelegt.

Lesen Sie dazu: Erklärung zu ASCII und UTF

Aufgabe HEX-Code entschlüsseln

Erinnern Sie sich noch? … bei Ceasar und Vigenere werden Buchstaben nach einer bestimmten Regel durch andere ersetzt. Eine Substitution wird auch bei UTF8 angewandt. Allerdings ist der Schlüssel (UTF8 Codiertabelle) allen zugänglich.. eine Verschlüsselung ist es nicht. Liste der UTF8 Codierung

```
\NormalTok{Verschlüsselung: UTF()}
\NormalTok{Key: https://www.utf8{-}chartable.de/}
\NormalTok{Verschlüsselter Text: 65 69 6e 65 73 63 68 69 63 6b 73}
\NormalTok{ 61 6c 68 61 66 74 65 77 65 6c 74 76 6f 6c 6c 65 72 6d}
\NormalTok{ 61 67 69 65 75 6e 64 64 75 6e 6b 6c 65 72 6d 61 65 63}
\NormalTok{ 68 74 65}

\NormalTok{Wie lautet der entschlüsselte Text?}
```

Lösung

eineschicksalhafteweltvollermagieunddunklermaechte

Digitale Verschlüsselung mit XOR

Es wird Zeit für ein erstes digitales Verschlüsselungsverfahren. Lesen und lösen Sie dazu:

Umrechung HEX to BIN herunterladen

Notebook zu XOR herunterladen

Verfahren der digitalen Verschlüsselung

Die Techniken der digitalen Verschlüsselung lassen sich unter *Transformation* als Oberbegriff zusammenfassen.

Eine Transformation kann aus diesens Schritten bestehen:

- **Substitution** : Ersetzen von Symbolen durch andere Symbole
- **Transposition/Permutation** : Vertauschung von Zeichen oder Bits
- **Diffusion** : Streuung der Klartext-Information über viele Bits im Geheimtext
- **Konfusion** : Komplexe, nichtlineare Beziehung zwischen Schlüssel und Geheimtext
- **Rundensystem** : Mehrfache Wiederholung von Substitution und Permutation
- **Key Schedule** : Algorithmus zur Ableitung von Rundenschlüsseln :

- **Blockchiffre** | Verschlüsselt Daten blockweise (z. B. 128 Bit bei AES)
- **Stromchiffre** | Verschlüsselt Bit für Bit oder Byte für Byte

Substitution

- UTF8 wendet Substituation an, es ersetzt anhand der UTF8 Code-Tabled ein Buchstabe durch eine Bitfolge
- **XOR** wendet Substitution an, es ersetzt die Bits auf Grund des Schlüssel durch andere Bits.

Transposition/Permutation

A = 01000001 key = 3

Meine eigene Regel:

- Buchstabencodierung UTF8, nur 7-Bit Buchstaben.
- Rechtes, fehlende Bit mit 1 auffüllen
- Teile die Bitfolge in Gruppen von 3 bit. Ergänze die Folge mit 1, falls es nicht aufgeht.
- Innerhalb jeder 3er-Gruppe vertauschen wir die Positionen nach Muster (1,2,3) -> (3,1,2)

```
\NormalTok{010 000 01 A}
\NormalTok{010 000 011  auffüllen}
\NormalTok{001 000 101 (1,2,3) {-}\textgreater{} (3,1,2)}

\NormalTok{Auflösen:}
\NormalTok{001 000 101 }
\NormalTok{010 000 011 (1,2,3) {-}\textgreater{} (3,2,1)}
\NormalTok{010 000 01  A}
```

Aufgabe Anwenden Transponieren

- Wenden Sie die obige Transposition auf einen 4 stellige Dezimale Zahl an.
- Wandeln Sie die Zifferen der Zahl mit UTF8 in eine binäre Zahlenfolge um. (ergibt 4 Bytes) Liste der UTF8 Codierung
- Transponieren Sie mit dem key=3 gemäss obiger Regel
- Tauschen Sie das Ergebnis mit Ihrem Lernpartner aus, findet er die ursprüngliche Zahl heraus?

Aufgabe Anwenden eines mehrfachen Transponieren

- ausprobieren: Permutation-Demo
- analysieren Sie die Schritte
- notieren Sie die Schritte auf.
- Codieren Sie einen von Ihnen gewählten Buchstaben.
  - Buchstabe mit UTF8 Tabelle in Bits umwandeln.
  - In der Permutations-Demo codieren, eigenen Schlüssel und Anzahl Runden wählen.
  - Mit Lernpartner Codierten Code, Schlüssel und Anzahl Runden austauschen.
  - Kann Ihr Lernpartner den richtigen Buchstaben herausfinden?

Anwendung in der Praxis mit Beispiel AES Verschlüsselung

AES Verfahren

AES ist ein modernes Verschlüsselungsverfahren. AES verschlüsselt, indem es den Klartext blockweise (128 Bit) in mehreren Runden mit dem Schlüssel verarbeitet. In jeder Runde passieren vier Schritte:

- Substitution: SubBytes -> jedes Byte wird durch die S-Box ersetzt
- Permutation: ShiftRows -> die Zeilen werden verschoben
- Diffusion: MixColumns -> die Spalten werden gemischt.
- AddRoundKey → der Block wird mit dem Rundenschlüssel per XOR verknüpft.
- Nach 10 (12 / 14) Runden entsteht der Geheimtext.

AES Verfahren Demo

Das sind einige Schritte. Die *AWS web demo* zeigt das anschaulich.

- Link anklicken, Random wählen, die Schritte beobachten.
- Das Resultat auch wieder dekodieren.

Eine AES-Animation, die das Verfahren versucht zu verdeutlichen.

- Probieren Sie es aus.

AES selber anwenden

Laden Sie das Notebook herunter, um AES praktisch anzuwenden. Lesen und lösen Sie:

Notebook zu AES herunterladen