XOR-Verschlüsselung Schritt für Schritt

In diesem Jupyter Notebook wird die **XOR-Verschlüsselung** erklärt und angewendet:

1. XOR-Idee verstehen
2. **XOR von Hand** an einem einfachen Beispiel nachvollziehen
3. **XOR mit Python** programmieren
4. Klartext als **HEX** und **BIN** darstellen
5. Schlüssel als **BIN** zeigen
6. XOR anwenden und Ergebnis als **HEX** ausgeben

Alle Beispiele verwenden einfache Texte und kleine Schlüssel, damit man den Vorgang gut nachvollziehen kann.

Was ist XOR?

XOR ("exclusive or") ist eine **logische Verknüpfung** auf Bit-Ebene. Sie kennen die Addition von zwei Zahlen 1+2=3 XOR wird nicht auf zwei dezimale Zahlen sonderen auf zwei Bits angewandt: Bit A und Bit B. Beide können 0 oder 1 sein.

Regeln für XOR:

Bit-A	Bit-B	Ergebnis A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Und nun kann man XOR auch auf zwei Bitfolgen anwenden, wobei immer die beiden Bits an der gleichen Position verknüpft werden.

Beispiel:

```
    0100'0100  (Bitfolge A -> Buchstabe 'H')
XOR 0100'1011  (Bitfolge B -> Schlüssel Buchstabe 'K' )
--------
    0000'1111  (Ergebnis, das muss nicht einem Buchstaben
                entsprechen!)
```

XOR-Verschlüsselung von Hand (Beispiel)

- **Klartext:** INFORMATIK
- **Schlüssel:** 'BYTE'
- Kodierung UTF8 (ein Byte pro Buchstabe

1. Klartext in Binär umwandeln INFORMATIK = 01001001 01001110 01000110 01001111 01010010 01001101 01000001 01010100 01001001 01001011
2. Schlüssel in Binär umwandeln BYTE = 01000010 01011001 01010100 01000101
3. Klartext und Schlüssel aufreihen, Schlüssel wird über die Länge des Klartexts wiederholt: 01001001 01001110 01000110 01001111 01010010 01001101 01000001 01010100 01001001 01001011 01000010 01011001 01010100 01000101 01000010 01011001 01010100 01000101 01000010 01011001
4. XOR bitweise anwenden 00001011 00010111 00010010 00001010 00010000 00010100 00010101 00010001 00001011 00010010 ACHTUNG : Die Bytes des Ergebnisses entsprechen nicht unbedingt druckbaren Zeichen! Man notiert das Ergebnis bin oder kompakter in HEX: 0B 17 12 0A 10 14 15 11 0B 12

XOR-Entschüsselung von Hand (Beispiel)

Prinzip der Entschlüsselung: - Wenn man ein Bit mit einem Schlüsselbit per XOR verknüpft, kann man mit **demselben Schlüssel** wieder zurückrechnen. - Also: (Klartext XOR Schlüssel) XOR Schlüssel = Klartext.

- **Verschlüsselt:** 0B 17 12 0A 10 14 15 11 0B 12
- **Schlüssel:** 'BYTE'
- Kodierung UTF8 (ein Byte pro Buchstabe)

1. Verschlüsselten Text in Binär umwandeln 0B 17 12 0A 10 14 15 11 0B 12 = 00001011 00010111 00010010 00001010 00010000 00010100 00010101 00010001 00001011 00010010
2. Schlüssel in Binär umwandeln BYTE = 01000010 01011001 01010100 01000101
3. Verschlüsselten Text und Schlüssel aufreihen, Schlüssel wird über die Länge des verschlüsselten Texts wiederholt: 00001011 00010111 00010010 00001010 00010000 00010100 00010101 00010001 00001011 00010010 01000010 01011001 01010100 01000101 01000010 01011001 01010100 01000101 01000010 01011001
4. XOR bitweise anwenden 01001001 01001110 01000110 01001111 01010010 01001101 01000001 01010100 01001001 01001011
5. Binär in Klartext umwandeln 01001001 01001110 01000110 01001111 01010010 01001101 01000001 01010100 01001001 01001011 = INFORMATIK

Aufgabe: XOR-Entschlüsselung von Hand

- Nehmen Sie ein Wort mit 6-8 Buchstaben
- Notieren Sie es als Grossbuchstaben
- Wandeln Sie es in Binär um: UTF-8 Tabelle Hinweis, Sie können in der Tabelle die Ansicht auf *binary* umstellen.
- Nehmen Sie einen Schlüssel mit 2-4 Buchstaben
- Wandeln Sie den Schlüssel in Binär um
- Verschlüsseln Sie den Text mit dem Schlüssel per XOR
- Notieren Sie das Ergebnis in HEX

Geben Sie den verschlüsselten Text und den Schlüssel an eine andere Person weiter. Diese Person soll den Text per Hand wieder entschlüsseln.

Aufgabe: XOR-Verschlüsselung mit Python

Zuerst definieren wir ein paar **Hilfsfunktionen**, um Texte in Bytes, Hex und Binärdarstellungen umzuwandeln und XOR anzuwenden.

```
\KeywordTok{def}\NormalTok{ text\_to\_bytes(text:
}\BuiltInTok{str}\NormalTok{, encoding: }\BuiltInTok{str} \OperatorTok{=}
\StringTok{\textquotesingle{}utf{-}8\textquotesingle{}}\NormalTok{)
}\OperatorTok{{-}\textgreater{}} \BuiltInTok{bytes}\NormalTok{:}
    \CommentTok{"""Wandelt einen Text in eine Folge von einzelnen Bytes
(UTF{-}8) um."""}
    \ControlFlowTok{return}\NormalTok{ text.encode(encoding)}

\KeywordTok{def}\NormalTok{ bytes\_to\_hex(data:
}\BuiltInTok{bytes}\NormalTok{) }\OperatorTok{{-}\textgreater{}}
\BuiltInTok{str}\NormalTok{:}
    \CommentTok{"""Gibt eine Folge von Bytes als Hex{-}String zurück (z.B.
\textquotesingle{}48 45 4C\textquotesingle{})."""}
    \ControlFlowTok{return} \StringTok{\textquotesingle{} \textquotesingle{
}}\NormalTok{.join(}\SpecialStringTok{f\textquotesingle{}}\SpecialCharTok{\
{}\NormalTok{b}\SpecialCharTok{:02X\}}\SpecialStringTok{\textquotesingle{}}
\ControlFlowTok{for}\NormalTok{ b }\KeywordTok{in}\NormalTok{ data)}


\KeywordTok{def}\NormalTok{ bytes\_to\_bin(data:
}\BuiltInTok{bytes}\NormalTok{) }\OperatorTok{{-}\textgreater{}}
\BuiltInTok{str}\NormalTok{:}
    \CommentTok{"""Gibt Bytes als Binär{-}String zurück (z.B.
\textquotesingle{}01001000 01000101\textquotesingle{})."""}
    \ControlFlowTok{return} \StringTok{\textquotesingle{} \textquotesingle{
}}\NormalTok{.join(}\SpecialStringTok{f\textquotesingle{}}\SpecialCharTok{\
{}\NormalTok{b}\SpecialCharTok{:08b\}}\SpecialStringTok{\textquotesingle{}}
\ControlFlowTok{for}\NormalTok{ b }\KeywordTok{in}\NormalTok{ data)}
```

```
13
14
15 \KeywordTok{def}\NormalTok{ xor\_bytes(data:
}\BuiltInTok{bytes}\NormalTok{, key: }\BuiltInTok{bytes}\NormalTok{)
}\OperatorTok{{-}\textgreater{}} \BuiltInTok{bytes}\NormalTok{:}
16     \CommentTok{"""XORt eine Daten{-}Bytefolge mit einem Schlüssel (der ggf.
wiederholt wird)."""}
17     \ControlFlowTok{if} \KeywordTok{not}\NormalTok{ key:}
18         \ControlFlowTok{raise} \PreprocessorTok{ValueError}\NormalTok{(}\St
ringTok{\textquotesingle{}Schlüssel darf nicht leer
sein\textquotesingle{}}\NormalTok{)}
19 \NormalTok{    key\_len }\OperatorTok{=} \BuiltInTok{len}\NormalTok{(key)}
20     \ControlFlowTok{return} \BuiltInTok{bytes}\NormalTok{(b
}\OperatorTok{\^{}}\NormalTok{ key[i }\OperatorTok{\%}\NormalTok{ key\_len]
}\ControlFlowTok{for}\NormalTok{ i, b }\KeywordTok{in}
\BuiltInTok{enumerate}\NormalTok{(data))}
```

Anwenden:

Sie fangen mit einem einfachen Beispiel an: Klartext: ‘ERAGON’ Schlüssel: ‘S’ Wie lautet der verschlüsselte Text in HEX?

Umsetzung in Python: (nur die Ideen, nicht die komplette Lösung) Klartext als Variable Schlüssel als Variable

Klartext in Bytes umwandeln und ausgeben Ergebnis in HEX umwandeln und ausgeben Ergebnis in Binär umwandeln und ausgeben

Schlüssel in Bytes umwandeln und ausgeben Ergebnis in HEX umwandeln und ausgeben Ergebis in Binär umwandeln und ausgeben

Klartext und Schlüssel per XOR verknüpfen Ergebnis in Binär ausgeben Ergebnis in HEX ausgeben

```
1 \CommentTok{\# Programmieren Sie die obige XOR Verschlüsselung in Python.}
2 \BuiltInTok{print}\NormalTok{(}\StringTok{"Meine
Verschüsselung"}\NormalTok{)}
```

```
Meine Verschüsselung
```

Aufgabe: XOR-Entschlüsselung mit Python

Schaffen Sie es den Ablauf für die Entsschlüsselung zu definieren und umzusetzen? **Geben Sie alle Zwischenergebnisse, HEX, BIN, Text … aus.**

```
\CommentTok{\# Programmieren Sie die zur vorangehenden Aufgabe passend XOR
Entschlüsselung in Python.}
\BuiltInTok{print}\NormalTok{(}\StringTok{"Meine
Entschlüsselung"}\NormalTok{)}
```

```
Meine Entschlüsselung
```

Aufgabe: Eigenes Beispiel mit einem Schlüsselwort (mehrere Bytes)

Implementieren Sie eine eigenes Beispiel mit einem KEY aus mehreren Buchstaben. Die Methode xor_bytes wird den Schlüssel über den Text **zyklisch wiederholen**. **Geben Sie alle Zwischenergebnisse, HEX, BIN, Text … aus.**

```
\CommentTok{\#\# XOR{-}Verschlüsselung mit Python}
\BuiltInTok{print}\NormalTok{(}\StringTok{"Eingenes Beispiel"}\NormalTok{)}
```

```
Eingenes Beispiel
```

Aufgabe: Gegenseitig Verschlüsseln und Entschlüsseln

Tauschen Sie mit einem Partner einen Schlüssel aus. Verschlüsseln Sie einen Text und schicken Sie nur den verschlüsselten Text. Der Partner soll den Text mit dem Schlüssel wieder entschlüsseln. Geben Sie alle Zwischenergebnisse, HEX, BIN, Text … aus.

```
\CommentTok{\# Partenerarbeit}
\BuiltInTok{print}\NormalTok{(}\StringTok{"Partenerarbeit"}\NormalTok{)}
```

```
Partenerarbeit
```

Aufgabe: Unvollständiger Schlüssel

Idee:

- Bob und Anne haben eine mit XOR verschlüsselte Nachricht ausgetauscht.
- Eve hat die ganze verschlüsselte Nachricht abgefangen.
- Zudem hat Eve auf verbotenem Weg vom Schlüssel der gesammt Länge 5, die ersten 4 Bytes stehlen können.

Schaffen Sie es Eve zu unterstützen und die ganze Nachricht zu entschlüsseln?

- Verschlüsselte Nachricht (HEX): 1f040215000d0b16041c1d03030c1b1c1d11141e09001d08061c1a1f0
- Bekannter Teil des Schlüssels (UTF8), 4 von 5 Buchstaben, der letzte Buchstabe fehlt: **hmpa**

```
\CommentTok{\# hack the code}
\BuiltInTok{print}
```

```
<function print>
```