

14.1 XOR encryption step by step

Peter Rutschmann

12.11.2025

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

Was ist XOR?

Sie kennen die Addition von zwei Zahlen $1+2=3$

XOR ("exclusive or") ist eine **logische Verknüpfung** auf Bit-Ebene. XOR wird nicht auf zwei dezimale Zahlen sondern auf zwei Bits (zBps. Bit-A und Bit-B) angewandt 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

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

```

1 \NormalTok{Beispiel:}
2
3 \NormalTok{    0100\textquotesingle{}0100 (Bitfolge A {-}\textgreater{}{}}
\NormalTok{Buchstabe \textquotesingle{}H\textquotesingle{}))}
4 \NormalTok{XOR 0100\textquotesingle{}1011 (Bitfolge B {-}\textgreater{}{}}
\NormalTok{Schlüssel Buchstabe \textquotesingle{}K\textquotesingle{} )}
5 \NormalTok{{-}{-}{-}{-}{-}{-}{-}{-}{-}}
6 \NormalTok{    0000\textquotesingle{}1111 (Ergebnis, das muss nicht einem}
\NormalTok{Buchstaben entsprechen!)}

```

XOR-Verschlüsselung

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

1. Klartext *INFORMATIK* in Binär umwandeln

```
1 \NormalTok{ I N F O R M A  
T I K}  
2 \NormalTok{01001001 01001110 01000110 01001111 01010010 01001101  
01000001 01010100 01001001 01001011}
```

2. Schlüssel *BYTE* in Binär umwandeln

```
1 \NormalTok{      B          Y          T          E}
2 \NormalTok{{01000010 01011001 01010100 01000101}}
```

3. Klartext und Schlüssel aufreihen, Schlüssel wird über die Länge des Klartexts wiederholt:

1	\NormalTok{	I	N	F	O	R	M	A
2	T	I	K}					
3	\NormalTok{01001001	01001110	01000110	01001111	01010010	01001101		
4	01000001	01010100	01001001	01001011}				
5	\NormalTok{	B	Y	T	E	B	Y	T
6	E	B	Y}					
7	\NormalTok{01000010	01011001	01010100	01000101	01000010	01011001		
8	01010100	01000101	01000010	01011001}				

4. Bei den beiden obigen Bit-Muster XOR bitweise anwenden

[illegible]

```

4 \NormalTok{00001011 00010111 00010010 00001010 00010000 00010100
  00010101 00010001 00001011 00010010}
5 \NormalTok{  0B      17      12      0A      10      14      15
  11      0B      12}

```

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-Entschlüsselung

Prinzip der Entschlüsselung: - Man wendet auf den *verschlüsselten Text* den *gleichen Schlüssel* an. - Wenn man ein Bit des *verschlüsselten Texts* mit einem *Bit des Schlüssels* per *XOR verknüpft*, erhält man das Bit des Klartexts zurück. - 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

```

1 \NormalTok{  0B      17      12      0A      10      14      15
  11      0B      12}
2 \NormalTok{00001011 00010111 00010010 00001010 00010000 00010100
  00010101 00010001 00001011 00010010}

```

2. Schlüssel in Binär umwandeln

```

1 \NormalTok{  B      Y      T      E}
2 \NormalTok{01000010 01011001 01010100 01000101}

```

3. Verschlüsselten Text und Schlüssel aufreihen. Schlüssel wird über die Länge des verschlüsselten Texts wiederholt:

```

1 \NormalTok{  0B      17      12      0A      10      14      15
  11      0B      12}
2 \NormalTok{00001011 00010111 00010010 00001010 00010000 00010100
  00010101 00010001 00001011 00010010}
3 \NormalTok{  B      Y      T      E      B      Y      T
  E      B      Y}
4 \NormalTok{01000010 01011001 01010100 01000101 01000010 01011001
  01010100 01000101 01000010 01011001}

```

4. Bei den beiden obigen Bit-Muster bitweise XOR anwenden

5. Binär in Klartext umwandeln

Aufgabe: Entschlüsseln Sie die Nachricht von Hand

- Lösung.. nicht spicken :-)

4

Aufgabe: Verschlüsselte Nachricht austauschen

- Verschlüsseln Sie ein Wort mit 6 Buchstaben und einem Schlüssel mit 3 Buchstaben.
- Notieren Sie das Ergebnis in HEX.
- Geben Sie Ihrem Lernpartner den verschlüsselten Text und den Schlüssel weiter.
- Kann er den Klartext wiederherstellen?

XOR-Verschlüsselung mit Python

- Vorgegeben ist die **Hilfsfunktion** `xor_bytes`, die zwei Byte-Arrays XOR-verknüpft.
- Weiter zeigt das Beispiel, wie man eine Text zerlegt und in Hex- und Binärdarstellungen ausgibt.

```
1 \CommentTok{\# Starten Sie den Block einmal und schauen Sie die Ausgaben an}
2
3 \CommentTok{\# XOR{-}Funktion fuer zwei bytes{-}arrays definieren}
4 \KeywordTok{def}\NormalTok{ xor\_bytes(data:
5     }\BuiltInTok{bytes}\NormalTok{, key: }\BuiltInTok{bytes}\NormalTok{)}
6     }\OperatorTok{{-}}\textgreater{} }\BuiltInTok{bytes}\NormalTok{:}
7     \CommentTok{""XORt eine Daten{-}Bytefolge mit einem Schlüssel (der ggf.
8     wiederholt wird).""}
9     \ControlFlowTok{if} \KeywordTok{not}\NormalTok{ key:}
10    \ControlFlowTok{raise} \PreprocessorTok{ValueError}\NormalTok{({}\StringTok{{}}\St
11    ringTok{\textquotesingle}Schlüssel darf nicht leer
12    sein\textquotesingle{}}\NormalTok{)}}
13    \NormalTok{ key\_len }\OperatorTok{=}\BuiltInTok{len}\NormalTok{(key)}
14    \ControlFlowTok{return} \BuiltInTok{bytes}\NormalTok{(b
15    }\OperatorTok{{-}}^{}\NormalTok{ key[i] }\OperatorTok{%}\NormalTok{ key\_len]
16    }\ControlFlowTok{for}\NormalTok{ i, b }\KeywordTok{in}
17    \BuiltInTok{enumerate}\NormalTok{(data))}
18
19 \NormalTok{text }\OperatorTok{=}\StringTok{"HALLO"}
20 \BuiltInTok{print}\NormalTok{({}\StringTok{"text "}
21 \OperatorTok{+}\NormalTok{ text})}
22
23 \CommentTok{\# text als einzelne Buchstaben ausgeben}
24 \BuiltInTok{print}\NormalTok{({}\StringTok{"text als
25 Buchstaben:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join(text))}
26
27 \CommentTok{\# text in einzelne Bytes zerlegen}
28 \NormalTok{textInBytes }\OperatorTok{=}\NormalTok{text.
29 encode()\StringTok{"utf{-}8"}\NormalTok{)}
30
31 \CommentTok{\# textInBytes in HEX ausgeben}
```

```

21 \BuiltInTok{print}\NormalTok{({}\StringTok{"textInBytes in
HEX:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\
SpecialCharTok{\}\}\NormalTok{x}\SpecialCharTok{:02X}\}\}\SpecialStringTok{"}\
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ textInBytes))}

22
23 \CommentTok{\# textInBytes in BIN ausgeben}
24 \BuiltInTok{print}\NormalTok{({}\StringTok{"textInBytes in
BIN:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\
SpecialCharTok{\}\}\NormalTok{x}\SpecialCharTok{:08b}\}\}\SpecialStringTok{"}\
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ textInBytes))}

25
26 \CommentTok{\# textInBytes wieder zu Buchstaben zusammensetzen}
27 \BuiltInTok{print}\NormalTok{({}\StringTok{"textInBytes wieder zu
Text:"}\NormalTok{, textInBytes.decode()}\StringTok{"utf{-}8"}\NormalTok{)})}

28
29 \BuiltInTok{print}\NormalTok{({})}
30 \CommentTok{\#schluessel als Buchstaben oder direkt als Bytes definieren}
31 \NormalTok{schluessel }\OperatorTok{=} \StringTok{"ADE"}
32 \NormalTok{schluesselInBytes }\OperatorTok{=}
\BuiltInTok{bytes}\NormalTok{({}\BaseNTok{0x41}\NormalTok{,
}\BaseNTok{0x44}\NormalTok{, }\BaseNTok{0x45}\NormalTok{)})}
33 \BuiltInTok{print}\NormalTok{({}\StringTok{"schluessel als Buchstaben: "}\
\OperatorTok{+}\NormalTok{ schluessel)}

34
35 \CommentTok{\# schluesselInBytes in HEX ausgeben}
36 \BuiltInTok{print}\NormalTok{({}\StringTok{"schluessel in HEX:"}\NormalTok{,
}\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\SpecialCharTok{\}\}\
NormalTok{x}\SpecialCharTok{:02X}\}\}\SpecialStringTok{"}\
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ schluesselInBytes))}

37
38 \CommentTok{\# schluesselInBytes in BIN ausgeben}
39 \BuiltInTok{print}\NormalTok{({}\StringTok{"schluessel in BIN:"}\NormalTok{,
}\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\SpecialCharTok{\}\}\
NormalTok{x}\SpecialCharTok{:08b}\}\}\SpecialStringTok{"}\
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ schluesselInBytes))}

40
41 \BuiltInTok{print}\NormalTok{({})}
42 \CommentTok{\# XOR{-}Operation durchfuehren}
43 \NormalTok{xorResultAlsBytes }\OperatorTok{=} \NormalTok{ xor\_bytes(textInBytes, schluesselInBytes)}

44
45 \CommentTok{\# xorResultAlsBytes in BIN ausgeben}
46 \BuiltInTok{print}\NormalTok{({}\StringTok{"xorResultAlsBytes in
BIN:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\
SpecialCharTok{\}\}\NormalTok{x}\SpecialCharTok{:08b}\}\}\SpecialStringTok{"}\
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ xorResultAlsBytes))}

47
48 \CommentTok{\# xorResultAlsBytes in HEX ausgeben}

```

49

```
\BuiltInTok{print}\NormalTok{({}\StringTok{"xorResultAlsBytes in
HEX:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\
SpecialCharTok{\}\}\NormalTok{x}\SpecialCharTok{:02X}\}\}\SpecialStringTok{"}
\ControlFlowTok{for}\NormalTok{ x }\KeywordTok{in}\NormalTok{ {
xorResultAlsBytes))}
```

```
text HALLO
text als Buchstaben: H A L L O
textInBytes in HEX: 48 41 4C 4C 4F
textInBytes in BIN: 01001000 01000001 01001100 01001100 01001111
textInBytes wieder zu Text: HALLO
```

```
schluessel als Buchstaben: ADE
schluessel in HEX: 41 44 45
schluessel in BIN: 01000001 01000100 01000101
```

```
xorResultAlsBytes in BIN: 00001001 00000101 00001001 00001101 00001011
xorResultAlsBytes in HEX: 09 05 09 0D 0B
```

Aufgabe: XOR-Verschlüsselung mit Python selber anwenden

Sie fangen mit einem einfachen Beispiel an: Klartext: 'ERAGON' Schlüssel: 'SIR' Wie lautet der verschlüsselte Text in HEX?

Vorgehen für Umsetzung in Python:

- Klartext als Variable definieren
- Klartext als einzelne Buchstaben ausgeben
- Klartext in Bytes umwandeln und ausgeben
- Klartext_In_Bytes in HEX umwandeln und ausgeben
- Klartext_In_Bytes in Binär umwandeln und ausgeben
- Schlüssel als Variable definieren
- Schlüssel als einzelne Buchstaben ausgeben
- Schlüssel in Bytes umwandeln und ausgeben
- Schlüssel_In_Bytes in HEX umwandeln und ausgeben
- Schlüssel_In_Bytes in Binär umwandeln und ausgeben
- Klartext und Schlüssel per XOR verknüpfen
- XOR-Ergebnis in Binär ausgeben

- XOR-Ergebnis in HEX ausgeben

```
1 \CommentTok{\# Programmieren Sie das Beschriebene in Python.}
2 \BuiltInTok{print}\NormalTok{{}}\StringTok{"Meine
Verschüsselung"}\NormalTok{{}}
```

Meine Verschüsselung

Lösung.. nicht spicken :-)

```
1 \NormalTok{text = "ERAGON"}
2 \NormalTok{print("text " + text)}
3 \NormalTok{print("text als Buchstaben:", " ".join(text))}
4 \NormalTok{textInBytes = text.encode("utf{-}8")}
5 \NormalTok{print("textInBytes in HEX:", " ".join(f"\{x:02X\}" for x in
textInBytes))}
6 \NormalTok{print("textInBytes in BIN:", " ".join(f"\{x:08b\}" for x in
textInBytes))}
7
8 \NormalTok{print()}
9 \NormalTok{schluessel = "SIR"}
10 \NormalTok{print("schluessel " + schluessel)}
11 \NormalTok{print("schluessel als Buchstaben:", " ".join(schluessel))}
12 \NormalTok{schluesselInBytes = schluessel.encode("utf{-}8")}
13 \NormalTok{print("schluesselInBytes in HEX:", " ".join(f"\{x:02X\}" for x in
schluesselInBytes))}
14 \NormalTok{print("schluesselInBytes in BIN:", " ".join(f"\{x:08b\}" for x in
schluesselInBytes))}
15
16 \NormalTok{print()}
17 \NormalTok{\# XOR{-}Operation durchfuehren}
18 \NormalTok{xorResultAlsBytes = xor\_bytes(textInBytes, schluesselInBytes)}
19 \NormalTok{print("xorResultAlsBytes in BIN:", " ".join(f"\{x:08b\}" for x in
xorResultAlsBytes))}
20 \NormalTok{print("xorResultAlsBytes in HEX:", " ".join(f"\{x:02X\}" for x in
xorResultAlsBytes))}
```

Aufgabe: XOR-Entschlüsselung mit Python selber anwenden

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


```

1 \CommentTok{\# Programmieren Sie die zur vorangehenden Aufgabe passend XOR
  Entschlüsselung in Python.}
2 \BuiltInTok{print}\NormalTok{({}\StringTok{"Meine
  Entschlüsselung"}\NormalTok{)}}
3
4 \NormalTok{encryptionInBytes = bytes([0x16, 0x1B, 0x13, 0x14, 0x06, 0x1C])}
  \BuiltInTok{bytes}\NormalTok{({}\BaseNTok{0x16}\NormalTok{,
  }\BaseNTok{0x1B}\NormalTok{, }\BaseNTok{0x13}\NormalTok{,
  }\BaseNTok{0x14}\NormalTok{, }\BaseNTok{0x06}\NormalTok{,
  }\BaseNTok{0x1C}\NormalTok{)}}

```

Meine Entschlüsselung

Lösung.. nicht spicken :-)

```

1 \NormalTok{encryptionInBytes = bytes([0x16, 0x1B, 0x13, 0x14, 0x06, 0x1C])}
2
3 \NormalTok{print("encryptionInBytes in BIN:", " ".join(f"\x:08b\\" for x in
  encryptionInBytes))}
4
5 \NormalTok{schluessel = "SIR"}
6 \NormalTok{schluesselInBytes = schluessel.encode("utf{-}8")}
7 \NormalTok{print("schluesselInBytes in BIN:", " ".join(f"\x:08b\\" for x in
  schluesselInBytes))}
8
9 \NormalTok{decryptionInBytes = xor\_bytes(encryptionInBytes,
  schluesselInBytes)}
10 \NormalTok{print("decryptionInBytes in BIN:", " ".join(f"\x:08b\\" for x in
  decryptionInBytes))}
11 \NormalTok{print(decryptionInBytes.decode("utf{-}8"))}

```

Aufgabe: Eigenes Beispiel mit einem Schlüsselwort

Implementieren Sie eine eigenes Beispiel mit einem KEY aus mehreren Buchstaben.
Geben Sie alle Zwischenergebnisse, HEX, BIN, Text ... aus.

```

1 \CommentTok{\#\# XOR{-}Verschlüsselung mit Python}
2 \BuiltInTok{print}\NormalTok{({}\StringTok{"Eigenes Beispiel"}\NormalTok{)}}

```

Eigenes Beispiel

Aufgabe: Gegenseitig Verschlüsseln und Entschlüsseln

Tauschen Sie mit einem Partner einen Schlüssel aus. Verschlüsseln Sie einen Text und geben Sie Ihrem Partner das verschlüsselte Ergebniss als HEX-Bytes encryptionInBytes = bytes([0x16, 0x1B, Der Partner soll den Text mit dem Schlüssel wieder entschlüsseln. Geben Sie alle Zwischenergebnisse, HEX, BIN, Text ... aus.

```
1 \CommentTok{\# Partenerarbeit}
2 \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 gesamt Länge 5, die ersten 4 Bytes stehlen können.

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

- Verschlüsselte Nachricht (HEX): 1f040215000d0b16041c1d03030c1b1c1d11141e09001d08061c1a1f021a01031404000f1f050706
- Bekannter Teil des Schlüssels (UTF8), 4 von 5 Buchstaben, der letzte Buchstabe fehlt:
hmpa

```
1 \CommentTok{\# hack the code}
2 \NormalTok{encryptionInBytes }\OperatorTok{=}
  \BuiltInTok{bytes}\NormalTok{(.fromhex()}\StringTok{"1f040215000d0b16041c1d03
030c1b1c1d11141e09001d08061c1a1f021a01031404000f1f050706"}\NormalTok{{})}
3 \BuiltInTok{print}\NormalTok{({}\StringTok{"encryptionInBytes in
HEX:"}\NormalTok{, }\StringTok{" "}\NormalTok{.join()}\SpecialStringTok{f"}\
SpecialCharTok{\}\NormalTok{x}\SpecialCharTok{:02X}\}\SpecialStringTok{"}
\ControlFlowTok{for}\NormalTok{x }\KeywordTok{in}\NormalTok{ encryptionInBytes))}
```

encryptionInBytes in HEX: 1F 04 02 15 00 0D 0B 16 04 1C 1D 03 03 0C 1B 1C 1D 11 14 1E 09 00 1D 08 06 1C 1A 1F 02 1A 01 03 14 04 00 0F 1F 05 07 06