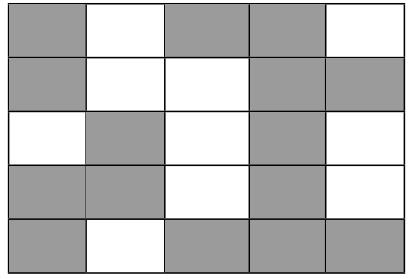
RICHELIEU

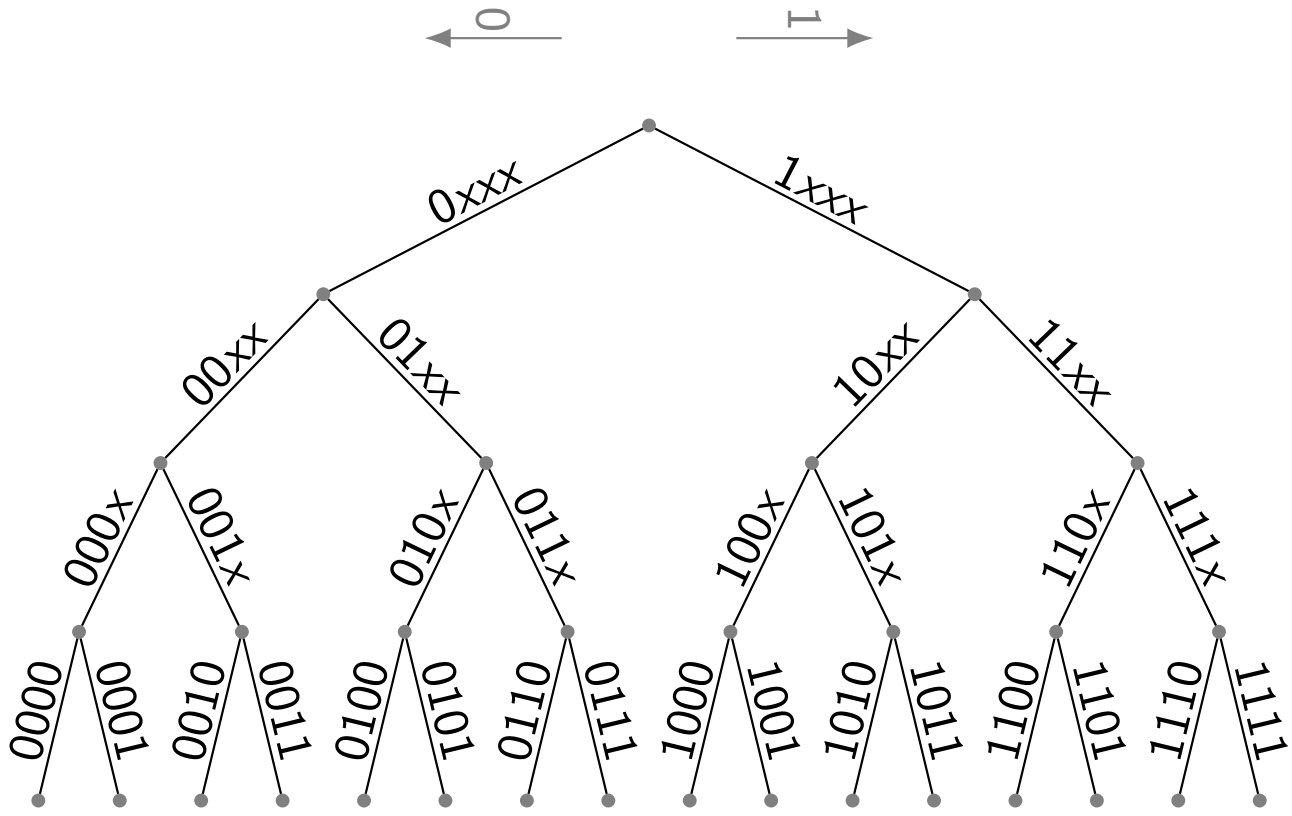
H	I	B	E	N
V	F	O	F	Q
R	E	M	T	A
N	O	T	L	I
U	K	G	A	N



	I			N
	F	O		
R		M		A
		T		I
	K			

Binäres Zahlenschloss

Zahlenschloss mit 4 Stellen. An jeder Stelle kann entweder eine 0 oder eine 1 gewählt werden.



Aufgabe RICHELIEU

Wie viele Schlüssel hat die Verschlüsslung von RICHELIEU bei einer Lochkarte von $n \times m$ Feldern?

Lösung Aufgabe RICHELIEU

Eine Lochkarte von $n \times m$ Feldern kann als binäres Zahlenschloss mit $n \cdot m$ vielen Stellen betrachtet werden. Jedes Feld kann entweder ausgeschnitten (0) oder nicht ausgeschnitten sein (1). Damit erhalten wir

$$2^{n \cdot m}$$

viele Schlüssel.

Kryptosystem mit vielen Schlüsseln

Beispiel: Der Text

YCXCBCWICISCYLIBVZRRBWCVCLQCKCBCT

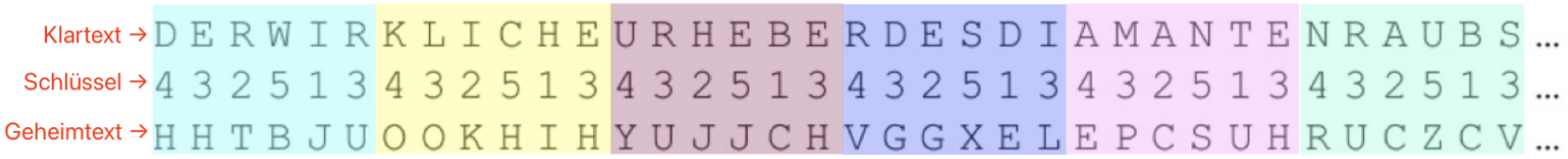
bedeutet:

GEHEDEINENWEGUNDLASSDIELEUTEREDEN

Wie viele Schlüssel gibt es?

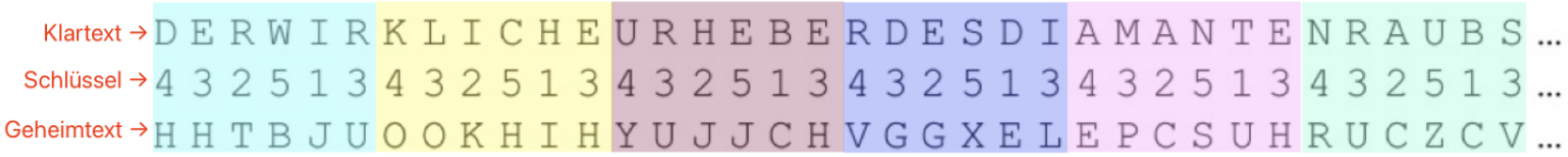
$$26! = 26 \cdot 25 \cdot 24 \cdot ... \cdot 3 \cdot 2 \cdot 1 = 4.06 \cdot 10^{26}$$

Jules Verne 📜



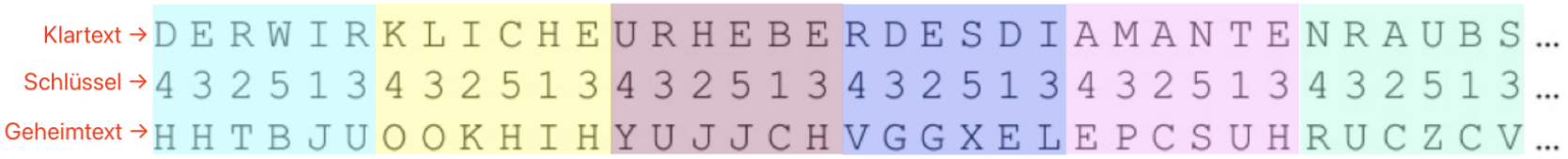
- ✅ Vorteil: nicht mehr monoalphabetische, sondern polyalphabetische Verschlüsselung

Jules Verne 📜



- ✅ Vorteil: nicht mehr monoalphabetische, sondern polyalphabetische Verschlüsselung
- ✅ Ein „E“ kann nun beispielsweise um 1 (F), 2 (G), 3 (H), 4 (I) oder 5 (J) Positionen verschoben sein.

Jules Verne 📜



- ✅ Vorteil: nicht mehr monoalphabetische, sondern polyalphabetische Verschlüsselung
- ✅ Ein „E“ kann nun beispielsweise um 1 (F), 2 (G), 3 (H), 4 (I) oder 5 (J) Positionen verschoben sein.
- 🤔 Problem: Nur 9 Schlüssel, einfach zu knacken bei bekannter Schlüssellänge...

Angriff auf Vigenère: Unbekannte Schlüssellänge

Kasiski-Test

🔑 Schlüssellänge unbekannt!

```
DER KLARTEXT WERDE GEHEIMTEXT
PLU TOPLUTOP LUTOP LUTOPLUTOP
SPL DZPCNXLI HYKRT RYASXXNXLI
```

- „NXLI“ kommt zweimal im Geheimtext vor
- Distanz zwischen NXLI = Vielfaches der Schlüssellänge

Angriff auf Vigenère: **Kasiski-Test**

Angenommen, wir haben folgenden Text, verschlüsselt mit dem Wort „BALMER“:

```
EINEFRAGEWIESEINESEINODERNICHTSEIN
BALMERBALMERBALMERBALMERBALMERBALM

HATINDERWELTDERQUANTENKEINEEINDEUTIGE
ERBALMERBALMERBALMERBALMERBALMERBALME

ANTWORT
RBALMER
```

Angriff auf Vigenère: **Kasiski-Test**

Wir suchen nun zuerst im Geheimtext nach Trigrammen, die mehrmals vorkommen:

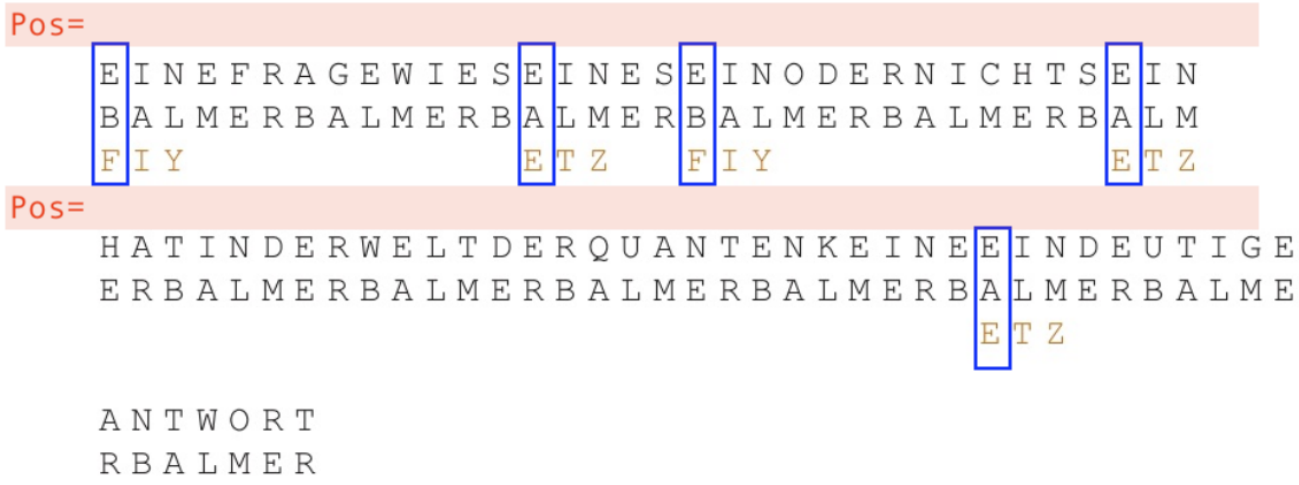
```
EINEFRAGEWIESEINESEINODERNICHTSEIN
BALMERBALMERBALMERBALMERBALMERBALM
FIY             ETZ   FIY             ETZ

HATINDERWELTDERQUANTENKEINEEINDEUTIGE
ERBALMERBALMERBALMERBALMERBALMERBALME
                                ETZ

ANTWORT
RBALMER
```

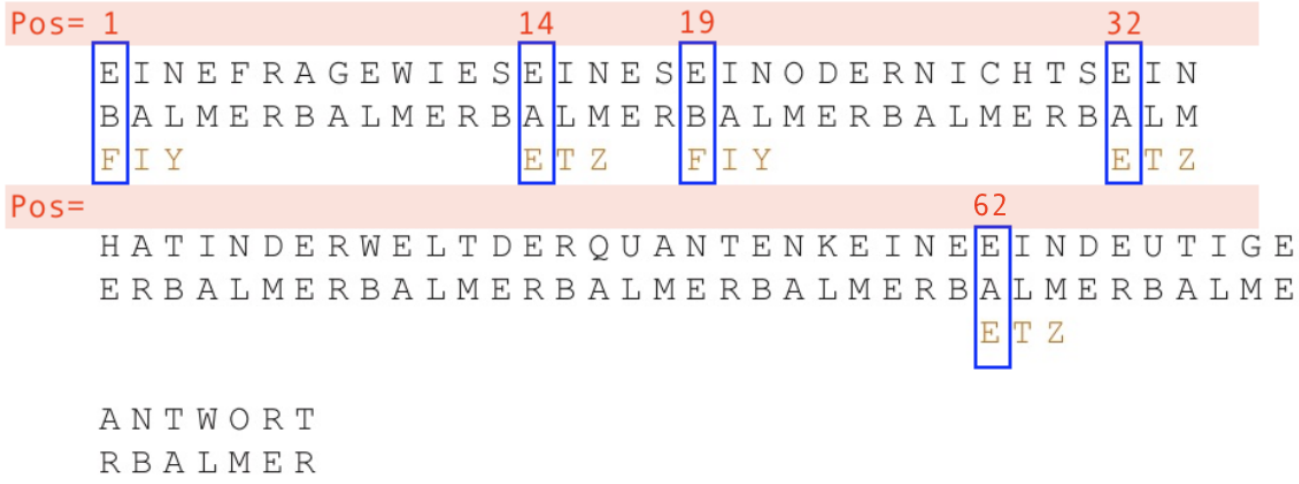
Angriff auf Vigenère: **Kasiski-Test**

Wir notieren die Anfangs-Positionen der Trigramme:



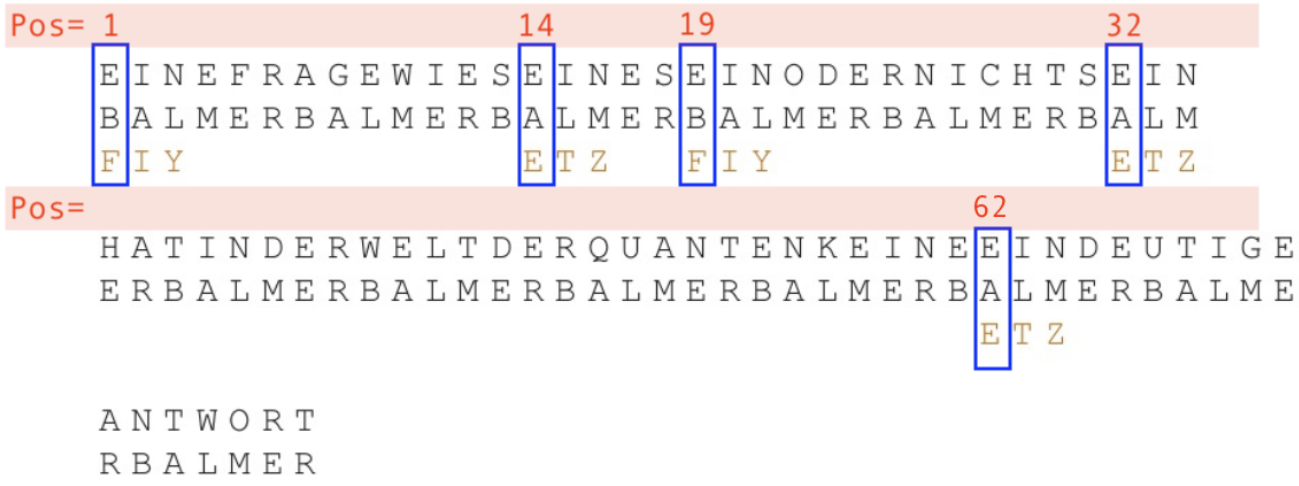
Angriff auf Vigenère: **Kasiski-Test**

Wir notieren die Anfangs-Positionen der Trigramme:



Angriff auf Vigenère: **Kasiski-Test**

Wir notieren die Anfangs-Positionen der Trigramme:



Nun können wir die Abstände zwischen den mehrfach vorkommenden Trigrammen berechnen:

$$62 - 14 = 48 = 2^4 \cdot 3$$

$$62 - 32 = 30 = 2 \cdot 3 \cdot 5$$

$$32 - 14 = 18 = 2 \cdot 3^2$$

GGT $= 6 \rightarrow$ Länge des Schlüsselworts.

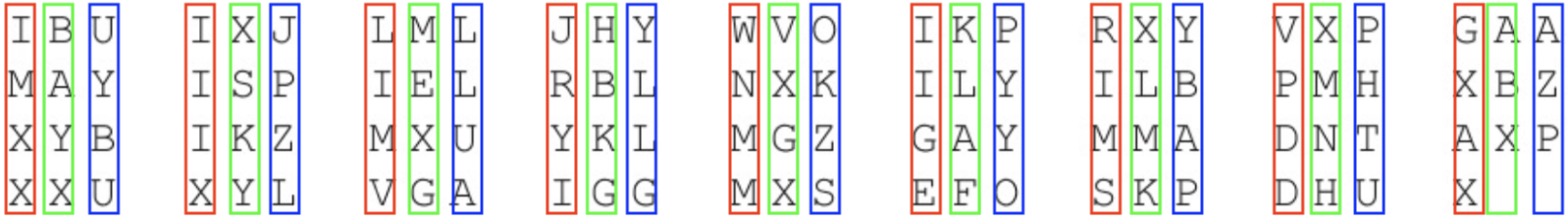
Angriff auf Vigenère: Bekannte Schlüssellänge

Beispiel: Geheimtext mit Schlüssellänge 3

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

Angriff auf Vigenère: Bekannte Schlüssellänge

Beispiel: Geheimtext mit Schlüssellänge 3

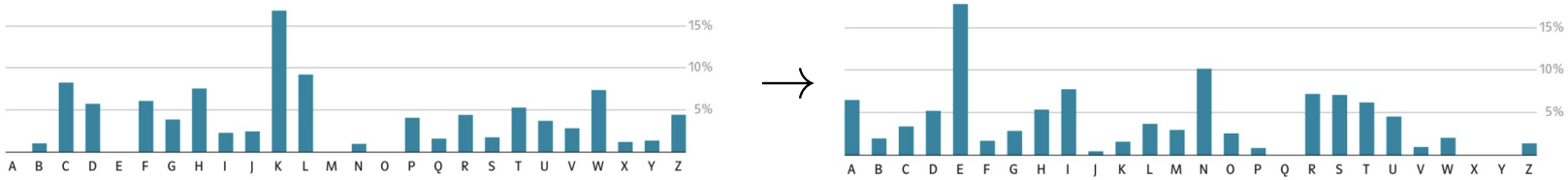


Angriff auf Vigenère: Bekannte Schlüssellänge

Beispiel: Geheimtext mit Schlüssellänge 3

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

Häufigkeitsanalyse pro Gruppe (rot, grün, blau):



Vigenère in Python

```
def caesar_buchstabe(buchstabe, verschiebung):
    return chr((ord(buchstabe) - ord("A") + verschiebung) % 26 + ord("A"))


def caesar(text, verschiebung):
    text_verschoben = ""

    for buchstabe in text:
        text_verschoben += caesar_buchstabe(buchstabe, verschiebung)

    return text_verschoben

def vigenere(text, schluessel, richtung="verschluesslung"):
    text_verschoben = ""
    k = 0
    for buchstabe in text:
        k %= len(schluessel)
        if richtung == "verschluesslung":
            text_verschoben += caesar_buchstabe(
                buchstabe, ord(schluessel[k]) - ord("A")
            )
        else:
            text_verschoben += caesar_buchstabe(
                buchstabe, -(ord(schluessel[k]) - ord("A"))
            )
        k += 1

    return text_verschoben
```