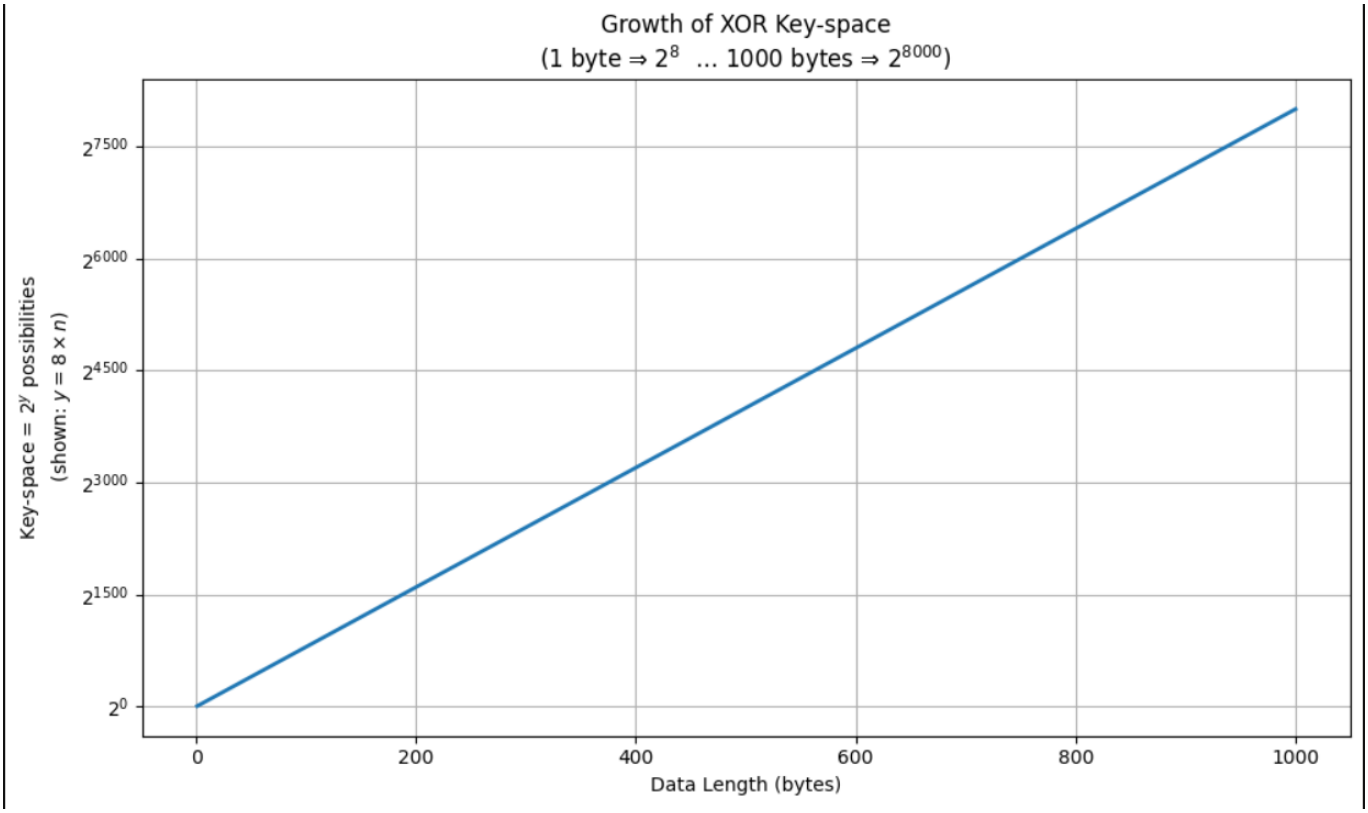
Security: XOR Vernam key space



xor_key_bruteforce.py

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.ticker import FuncFormatter

# 1) Prepare the data
data_lengths = np.arange(1, 1001)      # n = 1…1000 bytes
exponents    = 8 * data_lengths        # y = 8*n

# 2) Set up the formatter so ticks read as “2^{exponent}”
def exp_formatter(val, pos):
    # only label integer ticks
    iv = int(val)
    if abs(val - iv) < 1e-6:
        return f"$2^{{{notatki:iv}}}$"
    else:
        return ""

# 3) Plot
fig, ax = plt.subplots(figsize=(10,6))
ax.plot(data_lengths, exponents, linewidth=2)
ax.set_xlabel("Data Length (bytes)")
ax.set_ylabel("Key-space = $2^{y}$ possibilities\n    (shown: $y =
8\\times n$)")
```

```
ax.set_title("Growth of XOR Key-space\n(1 byte ⇒ $2^8$  … 1000 bytes ⇒
$2^{8000}$)")
ax.grid(True)

# apply our custom formatter
ax.yaxis.set_major_locator(plt.MaxNLocator(8))          # about 8 ticks
ax.yaxis.set_major_formatter(FuncFormatter(exp_formatter))

plt.tight_layout()
plt.show()
```