**Exercises – A3 og GDPR**

**Exercise 1**

The first thing we will do is to look at Cesarean rotation. I am not sure it was ever used, but it is good for making text which can't be read by the named, eye, but is easy to crack.

In the sample code:

1. **Find out how many letters to rotate the hidden message. Notice, the rotator can rotate both forwards and backwards (though that is not really necessary - why?)**

Det er ikke nødvendigt at gå baglæns, da modulus gør rotationen cirkulær. Det betyder at man kan gå fremad, f.eks. hvis en sætning er roteret med offset på 17, kan den afkodes ved at rotere den yderligere med et offset på 9 (9+17 = 26 som er længden på alfabetet). Antallet af bogstaver til rotationen er 9.

**Ascii-værdier:**

A = 65

Z = 90

a = 97

z = 122

**Et eksempel:**

N = 26

T = 84

T – A + offset + N (84 – 65 + 9 + 26) = 54

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse54 % N (54 % 26) = 2

2 + 65 = 67

Ascii-værdi: 67 = C

**Exercise 2**

This exercise uses an "incredible dumb and stupid" password scheme, as the actual input is a number between 1000 and 9999. You must find out which one. Your attack form here is brute force (trying them all).

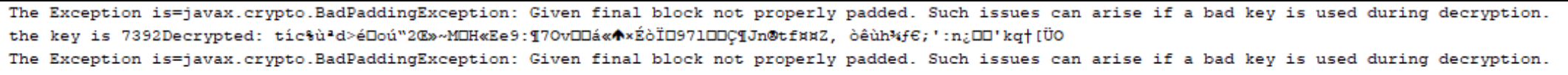
Hint: If you want to automate validation, the word "everything" occurs in the original text.

1. **Find out which input is used**

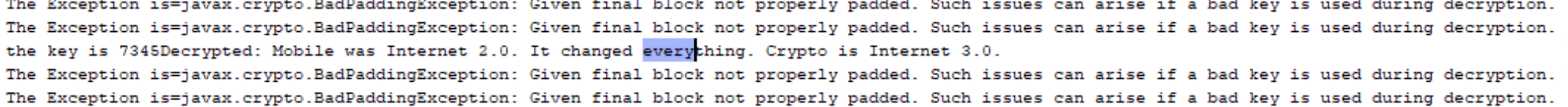
**Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse**

Nøglen skal være 16 bytes/128 bit, for kunne dekryptere beskeden. Det betyder at de nøgler der ikke er 16 bytes, vil give null eller en exception (”bad key”). Er nøglen dog 16 bytes, men den forkerte nøgle, vil beskeden der returneres være ulæselig/ubrugelig, men stadig være en besked.



Nøgle = passwordabc + 7345 (input)



**Exercise 3**

This exercise is about writing a small toy editor which allow you to store small texts in encrypted format. The save function is there, but you have to write the load.