

TI-Nspire CX II

2. Oefen met het gebruik van de caesar_ontsleutel module:
 - a. Ga naar de pagina met “**student_versleutel.py**”. Voer het programma uit om het platte tekstbericht te versleutelen. De gecodeerde tekst wordt automatisch opgeslagen op het besturingssysteem. Opmerking - Dit programma is gegeneraliseerd om een bericht te versleutelen met behulp van de ‘caesar_ontsleutel.py’ module aan het einde van het bestand. Modules helpen veelgebruikte code op te splitsen in een vorm die kan worden toegevoegd aan nieuwe programma's met de bijbehorende import verklaring.
 - b. Ga naar de pagina met “**student_ontsleutel.py**”. Voer het programma uit om de gecodeerde tekst die is opgeslagen op het besturingssysteem van het vorige programma te ontsleutelen. Probeer het platte tekstbericht te wijzigen en de twee programma's opnieuw uit te voeren.

3. Oefen met het gebruik van de karakterfrequentieanalyse:
 - a. Ga naar de pagina met ‘**chr_frequentie_oefenen.py**’ en voer het programma uit. Druk op de [var] toets, selecteer ‘code_to_chr()’ in het menu en voer een ASCII-code in als argument. Bijvoorbeeld, >>> code_to_chr(72) retourneert de letter “H”. Probeer een paar andere ASCII-codes om te zien welke karakters ze vertegenwoordigen.
 - b. Wanneer je het programma hebt uitgevoerd, werd de frequentie van elk teken in de gecodeerde tekst van de eerste alinea van het boek “*A Tale of Two Cities*” van Charles Dickens geteld. Twee lijsten bevatten de ASCII-code van elk teken in de tekst en de andere de frequentie van dat teken binnen de tekst. Opmerking: de sleutel is ingesteld op nul.
 - c. Ga naar de pagina met de gegevens- en statistiekengrafiek. De grafiek heeft de karaktercode (chr_code) op de horizontale as en de frequentie (f) op de verticale as. Selecteer [menu] -> 5: Window/Zoom -> 2: Zoom - Dit zal ervoor zorgen dat de gegevens in het venster passen.
 - d. Beweeg de cursor over het meest voorkomende teken. De ASCII-tekencode en frequentie worden (ASCII-code, frequentie) boven de cursor weergegeven. Beweeg vervolgens de cursor over het op één na meest voorkomende teken. Onthoud deze twee tekencodes voor de volgende stap.
 - e. Navigeer terug naar de Python-shell die werd aangemaakt bij het uitvoeren van het karaktertellen-oefenprogramma. Bij de Python REPL >>> prompt op die pagina, druk op de [var] toets, selecteer ‘code_to_chr()’ in het menu en voer een ASCII-tekencode in voor het meest voorkomende teken als argument voor de functie. Herhaal dit voor de tekencode van het op één na meest voorkomende teken.
 - f. Welke letter is het meest frequent in de tekst?
 - g. Navigeer terug naar de pagina met ‘**chr_frequentie_oefenen.py**’ en verander de sleutel naar drie, dat wil zeggen, sleutel=3. Herhaal de stappen “3.a” tot “3.d”. Merk op dat de ASCII-tekencodes voor het meest voorkomende en het op één na meest voorkomende teken zijn veranderd; ze zijn drie plaatsen naar rechts verschoven. Onthoud dat de ASCII-tekencode niet drie plaatsen is verschoven; in plaats daarvan is het teken in de chr_set lijst, die binnen het programma is gedefinieerd, drie posities in de lijst verschoven.

4. Verzenden van een versleuteld bericht:
 - o De **ontvanger**
 - o Selecteer ‘**student_ontvanger.py**’, verander de groep naar het toegewezen nummer en voer het programma uit **voordat** de zender zijn programma heeft uitgevoerd.

- De **zender**
 - Selecteer **'student_zender.py'**, bewerk de berichtstring, verander de groep naar het toegewezen nummer en voer vervolgens je programma uit nadat de ontvanger en hacker hun programma hebben gestart.
- De **hacker**
 - a. Selecteer **'student_hacker.py'**, verander de groep naar het toegewezen nummer en voer het programma uit **voordat** de zender zijn programma heeft uitgevoerd.
 - b. Na de man-in-the-middle-aanval die de gecodeerde tekst steelt, herhaal je het proces uit de oefening in stap 3 hierboven.
 1. Druk op de [var] toets en selecteer `count_ciphertext()`, druk opnieuw op de [var] toets en selecteer `'versleuteltekst.'` Het zou er als volgt uit moeten zien: `>>> tel_chrs_hack(versleuteltekst)`, druk vervolgens op enter om de tekens in het gestolen bericht te tellen.
 2. Ga naar de pagina met de gegevens- en statistiekengrafiek. Herhaal de analyse zoals beschreven in 3.c tot en met 3.e en 3.g. Opmerking: de versleutelsleutel is het aantal letters waarmee de platte tekst is verschoven in de gecodeerde tekst.
 3. Test je analyse van de sleutel door het `student_receiver` programma uit te voeren. Voer de gehackte sleutel in bij de prompt.
 4. Laat de `student_zender` het bericht opnieuw versturen. Zorg ervoor dat ze de sleutel niet aan de groep bekendmaken. Heb je het bericht gehackt?

De code

Zender

```

4.1 4.2 4.3 *3 - Cyber...ar! RAD 11/12
student_zender.py
from microbit_radio import *
from caesar_versleutel import *
# geheime sleutel moet zelfde zijn als ontvanger
bericht = "Experience keeps a dear school, but f
# sleutel moet een geheel getal zijn
sleutel = int(input("Geef de sleutel in: "))
kanaal = 1
groep = 1
versleuteldektst = versleutel(bericht,sleutel)
clear_history()
print("\nversleuteldektst = ",versleuteldektst)

```

Ontvanger

```

4.1 4.2 4.3 *3 - Cyber...ar! RAD 11/11
student Ontvanger.py
from microbit_radio import *
from caesar_versleutel import *
# geheime sleutel moet dezelfde zijn als zender
sleutel = int(input("Geef de sleutel in: "))
kanaal = 1
groep = 1
clear_history()
versleuteldektst = rx(kanaal,groep)
print("\nontvangen: ", versleuteldektst)
plattektst = ontsleutel(verseuteldektst,sleutel)
print("\nplattektst = ",plattektst)

```

Hacker

```

4.3 4.4 5.1 *3 - Cybers...ar! RAD 11/18
student_hacker.py
from microbit_radio import *
from caesar_cipher import *
from frequency_counter import *
chr_set = "ABCDEFGHJKLMNOPQRSTUVWXYZ"

def count_ciphertext(text):
    == count_chrs(text)

def code_to_chr(code):
    == return chr(code)

```

Extra uitdagingen

- Probeer een andere rol in je team.
- Verander de sleutel. Is de ontsleuteling hetzelfde?

Samengevat

- Versleutelingen worden gebruikt om platte tekstberichten te **verbergen** (verhullen) voor hackers.
- Een sleutel is vereist in het algoritme om de platte teksttekens naar de gecodeerde teksttekens te vertalen.
- De zender en ontvanger moeten dezelfde sleutel gebruiken.
- Frequentieanalyse is een techniek die gebruikt kan worden om berichten te ontsleutelen.

Tips voor als het misgaat

- Controleer of iedereen in het team hun toegewezen groepsnummer gebruikt.
- Zorg ervoor dat de ontvanger en hacker hun programma's uitvoeren en wachten voordat de zender het bericht verzendt.
- Zorg ervoor dat de zender en ontvanger dezelfde sleutel gebruiken en dat deze geheim blijft voor de hacker.