Cybersecurity 3: Hail, Caesar!
TI-84 Plus CE Python

Achtergrondinformatie

- Een substitutie**code** gebruikt dezelfde alfanumerieke- en interpunctietekens in zowel de platte tekst als de gecodeerde tekst.
- Om een bericht te **versleutelen**, wordt de alfabetische positie van elk platte tekstteken naar rechts of links verschoven (vertaald). Het teken op die nieuwe positie wordt het gecodeerde teken.
- Om een gecodeerd bericht te **ontsleutelen**, wordt het vertaalproces omgekeerd.
- Dezelfde sleutel moet worden gebruikt om een bericht te versleutelen en te ontsleutelen.
- De onderstaande tabel toont de posities van de tekens in het alfabet met twee interpunctietekens.

Karakter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	spatie	!
positie	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

- Voorbeeld: versleutel het karakter “H” in de platte tekst “HALLO” door gebruik te maken van een sleutel = 3 en een rechtse shift:
    1. Zoek ‘H’; het is het 8e teken in de tabel.
    2. Voeg de sleutel van 3 toe aan de positie; 8 + 3 = 11.
    3. Zoek positie 11; het teken is nu ‘K.’
    4. De eerste letter van de gecodeerde tekst is ‘K.’
    5. Als je het einde van de tabel bereikt, begin dan opnieuw te tellen vanaf het begin.
    6. Herhaal de vorige stappen voor de overige tekens; het resultaat is de gecodeerde tekst ‘KDOOR’.
- Een substitutiecode kan worden gekraakt door **een karakterfrequentieanalyse** van de gecodeerde tekst uit te voeren. In gewone platte tekst hebben de karakters van het alfabet gemiddeld een bepaalde frequentie in lange teksten. In de Engelse taal is een spatie (**ASCII 32**) het meest voorkomende teken en ‘E’ is de meest voorkomende letter. Aangezien ‘E’ de 5e letter in het alfabet is, kan de sleutel worden afgeleid door de frequentie van elk teken in de gecodeerde tekst te tellen. Bijvoorbeeld, als het meest frequente teken in de gecodeerde tekst ‘K’ is, dan is de kans groot dat ‘K’ (positie 11) wordt vervangen door ‘E’ (positie 5) en de verschuivingssleutel is 6, omdat 11 – 5 = 6. Dit type frequentieanalyse vereist een grote gecodeerde tekst om het meest frequente gecodeerde teken voor ‘E’ aan te nemen.

Wat is jouw opdract?

1. Oefen Caesar versleutelen:
    a. Gebruik een rechtse shift en sleutel = 6 samen met de bovenstaande tabel. Vul in de lege ruimte hieronder de versleuteldede tekst in.

       Platte tekst: HAIL CAESAR! Versleutelde tekst: _ _ _ _ _ _ _ _ _ _ _ _

    b. Gebruik een rechtse shift en sleutel = 6 samen met de bovenstaande tabel. Vul in de lege ruimte hieronder de ontsleutelde tekst in.

       Platte tekst: _ _ _ _ _ _ _ _ _ _ _ _ _ _ Versleutelde tekst: ICHKXEOYEL TF

    c. Sluit de micro:bit aan op de rekenmachine. Open het programma **‘CRYPT_3.py’** en bekijk de code van het programma. Voer het programma uit om je versleuteling uit 1a te controleren. Als je het programma uitvoert kun je met de **[var]**-knop op je rekenmachine de verschillende functies en variabelen uit het programma selecteren.
       Voor dit voorbeeld typ je: encipher(“HAIL CEASAR!”,6,chr_set_1). Komen de coderingstappen overeen met jouw stappen om de platte tekst in oefening 1a te versleutelen?
    d. Controleer ook je ontsleuteling van voorbeeld 1a. Voer nu in:
       encipher(“ICHKXEOYEL TF”,6,chr_set_1). Maak weer gebruik van de **[var]**-knop.

e. Wat gebeurt er met de gecodeerde tekst als je de sleutel naar een ander getal wijzigt en het programma opnieuw uitvoert?

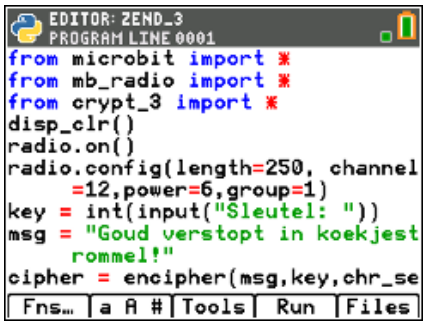
2. Oefen met het gebruik van de karakterfrequentieanalyse:
   a. Open het programma **'OEFEN_3'** en voer het uit. Het programma telt de frequentie van de karakters in eerste alinea van het boek *"A Tale of Two Cities" van Charles Dickens*.
   b. Verlaat Python en maak met behulp van StatPlot een histogram van L1 (de ascii-code) en L2 (de frequentie). Gebruik het Zoom-menu en daarin 9: ZoomStat om de assen passend te maken. Zoek met de *trace* optie en de pijltoetetsen het meest voorkomende en het op een na meest voorkomende karakter. Onthoud deze twee codes voor de volgende stap.
   c. Open nu in python het programma **'CHR_CT_3'** en voer het uit. Druk op de [var]-knop en selecteer get_chr(). Vul tussen de haakjes de codes uit de vorige stap in om de bijbehorende tekens te vinden.
   d. Welke letter is het meest frequent in de tekst?

3. Verzenden van een versleuteld bericht:
   - Controleer dat alle groepsleden hetzelfde groepsnummer gebruiken.
   - De **ontvanger**
     - Open het programma **'ONTV_3.py'**. Dit programma importeerd de functies uit CRYPT_3.py en gebruikt chr_set_2. Dus ook kleine letters, cijfers en extra tekens zijn toegestaan. Pas het groepsnummer aan en run het programma voor dat de zender zijn programma uitvooert.
   - De **zender**
     - Open het programma **'ZEND_3.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. Open **'HACK.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_chr(), druk opnieuw op de [var] toets en selecteer 'stolen_msg' Het zou er als volgt uit moeten zien:
           >>> count_chrs (stolen_msg), druk vervolgens op enter om de tekens in het gestolen bericht te tellen.
        2. Sluit Python en maak een StatPlot van L1(Ascii-codes) en L2(frequentie). Gebruik nu Trace om de tekens met de hoogste frequentie op te sporen.
        3. Ga weer naar Python en bepaal met het programma **'CHR_CT_3'** de tekens welke bij die ascii-codes horen. Hierme kun je de sleutel raden.
        4. Test je analyse van de sleutel door het student_receiver programma uit te voeren. Voer de gehackte sleutel in bij de prompt. Laat de zender het bericht opnieuw versturen. Zorg ervoor dat ze de sleutel niet aan de groep bekendmaken. Heb je het bericht gehackt?

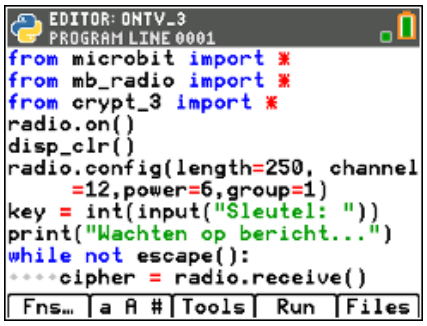
Cybersecurity 3: Hail, Caesar!
TI-84 Plus CE Python

De code

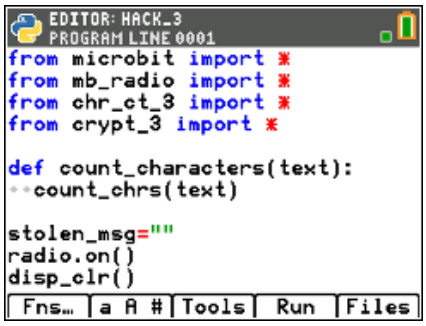
Zender



Ontvanger



Hacker



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.

Cybersecurity 3: Hail, Caesar!
TI-84 Plus CE Python

Bestanden

- Zet de onderstaande programma's op je rekenmachine m.b.v. de TI Connect CE software. De link om deze software te downloaden staat hier. Telkens als je met een nieuw onderdeel begint kun je het beste eerst de gebruikte programma's wissen en daarna de programma's voor het nieuwe ondedeel weer op je rekenmachine zetten.

Naam	Beschrijving
OEFEN_3.py	Oefenen met het maken van een frequentie historam van gecodeerde tekst.
ZEND_3.py	Zenden van een tekstbericht naar een ontvanger met Ceasar-versleuteling.
ONTV_3.py	Ontvangen van een tekstbericht met Ceasar-versleuteling.
HACK_3.py	Ontvangen van een versleuteld bericht.
CRYPT_3.py	Hulpprogramma met de functies voor de Ceasar-versleuteling
CHR_CT_3.py	Hulpprogramma voor het bepalen van de frequentie van de tekens in een tekst.