So funktioniert es

- Bei einer Substitutions-Chiffrierung werden sowohl im Klartext als auch im Chiffretext dieselben alphanumerischen Zeichen und Interpunktionszeichen verwendet. Bei der Verschlüsselung einer Nachricht werden die Klartextzeichen nach rechts oder links an eine neue Position in der chiffrierten Nachricht verschoben (übersetzt).
- Der Schlüssel ist die Anzahl der Positionen in einer Zeichentabelle, auf die die Klartextzeichen übersetzt werden. Zum Verschlüsseln und Entschlüsseln einer Nachricht muss derselbe Schlüssel verwendet werden.
- Die folgende Tabelle zeigt die Position der Großbuchstaben im Alphabet und zwei Satzzeichen.

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	!	Leer
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

- Beispiel: Chiffrierung der Klartextnachricht "HOWDY" mit einem Schlüssel = 3 und einer Rechtsverschiebung:
    1. Suchen Sie das "H"; es ist das 8. Zeichen in der Tabelle.
    2. Addieren Sie den Schlüssel 3 zu der Position: 8 + 3 = 11
    3. Suchen Sie Position 11: es ist das Zeichen "K".
    4. Der erste Buchstabe des Chiffretextes ist "K".
    5. Wenn Sie am Ende des Tabelle angekommen sind, zählen Sie von Anfang an weiter.
    6. Wiederholen Sie die vorangegangenen Schritte für die übrigen Zeichen; das Ergebnis ist der Chiffretext "KRZGA".

Was ist zu tun?

1. Üben Sie die Caesar-Chiffrierung:
    a. mit einer Rechtsverschiebung und Schlüssel = 6. Schreiben Sie den Chiffriertext auf Ihr Papier.
    Klartext: AVE CAESAR! Chiffretext: _ _ _ _ _ _ _ _ _ _ _ _
    b. mit einer Rechtsverschiebung und Schlüssel = 6. Schreiben Sie den Klartext auf Ihr Papier.
    Klartext: _ _ _ _ _ _ _ _ _ _ _ _ _ _ Chiffriertext: G!KEIGKYGXF
    c. Öffnen Sie die Datei *CyberSecurity – Ave Caesar.tns*
    d. Rufen Sie in der Datei die Seite "*check_practice_1.py*" auf und führen Sie das Programm aus, um Ihre Verschlüsselung zu überprüfen. Lesen Sie die Kommentare im Python-Code, damit Sie verstehen, was der Code macht. Stimmen die Codierungsschritte mit Ihren Schritten zur Verschlüsselung des Klartextes in Übung 1 überein?
    e. Rufen Sie in der Datei die Seite "*check_practice_2.py*" auf und führen Sie das Programm aus, um Ihre Entschlüsselung zu überprüfen. Lesen Sie die Kommentare im Python-Code, damit Sie verstehen, was der Code macht. Stimmen die Codierungsschritte mit den Schritten überein, die Sie zur Entschlüsselung des Chiffriertextes in Übung 2 durchgeführt haben? Was passiert mit der Nachricht, wenn Sie den Schlüssel auf eine andere Zahl ändern und das Programm erneut ausführen?
2. Üben Sie die Verwendung des Moduls *caesar_cipher*:
    a. Wechseln Sie in der Datei auf die Seite "*student_encipher.py*". Führen Sie das Programm aus, um die Klartextnachricht zu verschlüsseln. Der Chiffriertext wird automatisch im Betriebssystem gespeichert. Hinweis: Dieses Programm wurde verallgemeinert, um eine Nachricht zu verschlüsseln, indem das Modul "*caesar_cipher.py*" zu Beginn des Codes importiert wird.

Module helfen dabei, häufig verwendeten Code in eine Form zu bringen, die mit der Anweisung *import* zu neuen Programmen hinzugefügt werden kann.

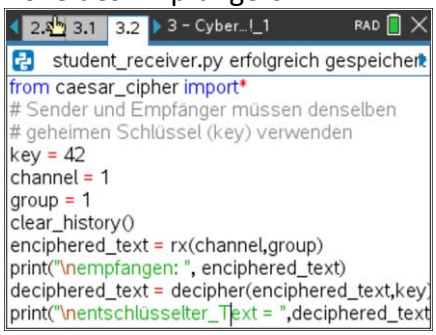
b. Wechseln Sie in der Datei auf die Seite "*student_decipher.py*". Führen Sie das Programm aus, um den im Betriebssystem gespeicherten Chiffriertext des vorherigen Programms zu entschlüsseln. Versuchen Sie, die Klartextnachricht zu ändern und beide Programme erneut auszuführen.

3. Senden einer verschlüsselten Nachricht:
   - Der *Empfänger*
     - wechselt auf die Seite "*student_receiver.py*", ändert den Kanal auf die zugewiesene Nummer und führt das Programm aus**, bevor** der *Sender* sein Programm ausgeführt hat.
   - Der *Sender*
     - wechselt auf die Seite "*student_sender.py*", ändert den Kanal auf die zugewiesene Nummer, bearbeitet dann die Nachrichtenzeichenfolge und führt das Programm aus**, nachdem** der *Empfänger* und der *Hacker* ihre Programme gestartet haben.
   - Der *Hacker*
     - wechselt auf die Seite "*student_hacker.py*", ändert den Kanal auf die zugewiesene Nummer, und führt das Programm aus**, bevor** der *Sender* sein Programm ausgeführt hat.
   - Nachdem das Team die Aktivität ausgeführt hat, sollte der *Sender* nur die Nachricht und den Schlüssel ändern und den Schlüssel an den *Empfänger* weitergeben. Teilen Sie dem *Hacker* den neuen Schlüssel nicht mit; **halten Sie ihn geheim**! Kann der *Hacker* Ihre Nachricht im Klartext lesen, wie er es bei der Aktivität "Klartext" getan hat?

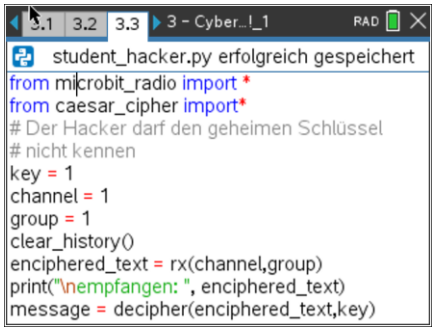
Die Programme

Rolle des Senders

Rolle des Empfängers



Rolle des Hackers



Weitere Übungen

- Wechseln Sie in Ihrem Team die Rollen und versuchen Sie es erneut.
- Wiederholen Sie die Aktivität mit anderen Gruppennummern. Wie wäre es, wenn Sie die gleiche Nummer wie ein anderes Team im Raum verwenden?

Prüfen Sie Ihr Verständnis

- Chiffren werden verwendet, um Nachrichten im Klartext vor *Hackern* **zu verschleiern** (zu verbergen).
- Ein Schlüssel ist erforderlich, um die Klartextzeichen in die Chiffrezeichen zu übersetzen.
- *Sender* und *Empfänger* müssen denselben Schlüssel verwenden.

Hilfe

- Vergewissern Sie sich, dass alle Teammitglieder die ihnen zugewiesene Gruppennummer verwenden.
- Stellen Sie sicher, dass der *Empfänger* und der *Hacker* ihre Programme ausführen und warten, bis der *Sender* die Nachricht übertragen hat.
- *Empfänger* und *Hacker* können ihre Programme bei Bedarf durch Drücken der <esc> Taste jederzeit beenden.
- Vergewissern Sie sich, dass *Sender* und *Empfänger* denselben Schlüssel verwenden und dieser vor Hackern geschützt ist.