IoT Security: labo 3 - encryptie/decryptie

Oefening 1: versleutelen met de Caesar versleuteling

Het Caesar Cipher is een van de eenvoudigste en bekendste encryptietechnieken. Het is een soort substitutieversleuteling waarbij elke letter in de plaintext (de te versleutelen tekst) wordt 'verschoven' een bepaald aantal plaatsen verderop in het alfabet. Dit aantal plaatsen, bekend als de 'sleutel', is hetzelfde voor alle letters in de bericht.

Bijvoorbeeld, met een verschuiving (of sleutel) van 3:

- A wordt D
- B wordt E
- C wordt F
- ...
- Z wordt C

De algemene formule om een letter te encrypteren in de Caesar Cipher is: [E_n(x) = (x + n) \mod {26}]

Waarbij:

- (E_n(x)) de uitvoer is na encryptie,
- (x) de originele letter is (uitgedrukt als een getal, waarbij A=0, B=1, enzovoort),
- (n) de verschuiving (of sleutel),
- en (\mod {26}) de modulo-operator die ervoor zorgt dat de encryptie 'rondgaat' na Z.

De decryptie werkt op dezelfde manier, alleen gebruik je dan de tegengestelde verschuiving: [D_n(x) = (x - n) \mod {26}]

Het Caesar Cipher is zeer gemakkelijk te breken en daarom in de hedendaagse cryptografie niet meer gebruikt voor beveiligingstoepassingen, maar het wordt wel vaak aangehaald als een eenvoudig en begrijpelijk voorbeeld bij de introductie tot cryptografie.

Jouw opdracht: je schrijft een programma dat in staat is een boodschap te encrypteren met de Caesar versleuteling en de boodschap ook te decrypteren. Je hoeft geen interface te maken: werk met variabelen. Variabelen voor boodschap, sleutel, actie: encrypteren/decrypteren.

Oefening 2: het bruteforcen van een met de Caesar versleuteling geëncrypteerde boodschap

Jouw opdracht: je gaat een met Caesar versleuteling geëncrypteerde boodschap decrypteren. Je kent de sleutel niet. Je gaat dus voor elke mogelijke sleutel een decryptie tonen. Je logt deze naar de console. Je zal visueel snel kunnen zien welke sleutel de echte vertaalde boodschap heeft teruggegeven. En dat volstaat hier.

Oefening 3: kolomtranspositie versleuteling

Het Caesar cipher is relatief eenvoudig te hacken. Dat heb je zelf kunnen vaststellen in de vorige opdracht. Om onze boodschappen beter te beveiligen hebben we een betere encryptie nodig. In deze oefening gaan

we gebruik maken van kolomtranspositie versleuteling.

kolomtranspositie werkt als volgt:

1. De te versleutelen boodschap (plaintext) wordt horizontaal geschreven in een vastgesteld aantal kolommen. Het aantal kolommen is typisch gebaseerd op een sleutelwoord of een getal.
2. De boodschap wordt vervolgens verticaal uitgelezen, kolom voor kolom, om de gecodeerde boodschap (de cipher tekst) te creëren.
3. Stel, we hebben de tekst "HELLO WORLD" en we kiezen om deze in drie kolommen te schrijven:

```
H E L
L O W
O R L
D
```

Vervolgens lezen we de tekst verticaal per kolom: "HLOEDLORW".

4. Om te decrypten wordt de proces omgekeerd. De cipher tekst wordt geschreven in een aantal kolommen dat wordt bepaald door de lengte van de originele boodschap en de sleutel, en vervolgens horizontaal uitgelezen.

Merk op: hoewel de kolomtranspositie cipher veel sterker is dan eenvoudige substitutieciphers vanwege het herarrangeren van de letters, kan het nog steeds effectief worden gebroken met genoeg tekst en analyse. Het is echter een goed voorbeeld van hoe transpositie als een cryptografisch mechanisme werkt.

Jouw opdracht: schrijf een programma dat encrypteert met deze versleuteling.

Je hoeft geen interface te maken: werk met variabelen. Variabelen voor boodschap, sleutel. Je logt de geëncrypteerde boodschap naar de terminal.

Oefening 4: kolomtranspositie decryptie

Jouw opdracht: schrijf een programma dat decrypteert met deze versleuteling.

Je hoeft geen interface te maken: werk met variabelen. Variabelen voor boodschap, sleutel. Je logt de gedecrypteerde boodschap naar de terminal.

Oefening 5: encrypteren en decrypteren van bestanden

Je breidt oefeningen 3 en 4 uit (importeer hen 😉) en zorg ervoor dat er van een bronbestand kan worden ingelezen en naar een nieuw bestand kan worden geschreven.

Je hoeft geen interface te maken: werk met variabelen. Je logt basisinfo naar de terminal. Meet hoelang de encryptie/decryptie duurde en log naar het scherm.

Oefening 6: detecteer of een string Nederlandse taal bevat

Op basis van een tekstbestand met Nederlandse woorden probeer je na te gaan of een string geschreven is in Nederlands of niet. Je kan deze functie testen op oefening 2 of je schrijft een extra script dat je programma werkt.

Het bestand in kwestie vind je op Digitap: woordenlijst.txt

Oefening 7: het bruteforcen van een met de kolomtranspositie versleuteling geëncrypteerde boodschap

Voor de doorzetters 😉 Probeer nu tot slot een kolomtranspositie geëncrypteerde string te bruteforce decrypteren. Je kent uiteraard de key niet, dus je overloopt ze allemaal. Maar je probeert dmv het herkennen van het Nederlands uit oefening 6 enkel de juiste oplossing te weerhouden.