IoT Security: labo 4 - encryptie/decryptie deel 2

Oefening 8: vermenigvuldigingsversleuteling

In de context van cryptografie verwijst vermenigvuldigingsversleuteling specifiek naar een soort substitutiecijfer waarbij elke letter in het bericht wordt vervangen door een andere letter. De vervanging vindt plaats volgens een vooraf bepaalde wiskundige formule, die een vermenigvuldiging bevat als centrale bewerking. Vaak wordt bij klassieke cijfers gewerkt met de numerieke equivalenten van letters (bijvoorbeeld a=0, b=1, ..., z=25) en wiskundige bewerkingen worden in een modulo stelsel uitgevoerd om binnen het alfabet te blijven. In het geval van een multiplicative cipher, wordt elk getal (dat een letter voorstelt) vermenigvuldigd met een gekozen sleutel en vervolgens gemoduleerd door de lengte van het alfabet om een nieuw getal te krijgen dat dan wordt omgezet naar een letter, resulterend in de versleutelde tekst.

- Je zoekt op hoe vermenigvuldigingsversleuteling werkt
- Welke toepassingen kent/kende dit soort versleuteling?
- Je schrijft een Python-script dat volgens deze versleuteling encrypteert en decrypteert.

Tip: niet alle sleutels kunnen gebruikt worden

Bij het gebruik van het vermenigvuldigingscijfer voor versleuteling kunnen niet alle sleutels worden gebruikt. Een sleutel is geldig als en slechts als deze een wederzijds priemgetal is met betrekking tot het aantal gebruikte symbolen. Dus, als we een alfabet gebruiken van 26 letters, dan moet de sleutel die we kiezen voor het vermenigvuldigingscijfer een getal zijn dat geen gemeenschappelijke factoren heeft met 26, behalve 1 (dat wil zeggen, het moet copriem zijn met 26). Dit is essentieel om ervoor te zorgen dat de vermenigvuldigingsversleuteling omkeerbaar is, wat betekent dat de originele tekst altijd kan worden hersteld door te ontcijferen. In de context van een alfabet van 26 letters, zijn geldige sleutels bijvoorbeeld 3, 5, 7, enz., omdat deze getallen copriem zijn met 26. Sleutels als 2, 4 of 6 zijn niet bruikbaar omdat ze gemeenschappelijke factoren hebben met 26, waardoor de versleuteling niet omkeerbaar zou zijn.

```
def ggd(a, b):
# algoritme van Euclides
    while a != 0:
        a, b = b % a, a
    return b


if ggd(sleutel, len(SYMBOLS)) == 1:
     # sleutel is valide!
```

Oefening 9: de vermenigvuldigigingsversleuteling bruteforcen

Jouw opdracht: je gaat een met vermenigvuldigversleuteling geëncrypteerde boodschap decrypteren. Je kent de sleutel niet. Je gaat dus voor elke mogelijke sleutel een decryptie tonen. Je maakt gebruikt van oefening 6 om automatisch de juiste gedecrypteerde boodschap te vinden.

Oefening 10: "Affiene Versleuteling" of "Affiene Cijfer"

Affiene versleuteling combineert vermenigvuldigigingsversleuteling en Caesar versleuteling. Het woord "affiene" (in het Engels spreken we over Affine cipher) refereert naar het soort wiskundige functies dat gebruikt wordt (affiene transformaties). Een affiene functie is in de wiskunde een functie die een lijn of een deel ervan representeert (elk punt wordt verplaatst langs een constante lijn of vector), en dat is precies wat er gebeurt bij deze versleutelingsmethode: elke letter of symbool wordt langs een zekere "afstand" (vastgesteld door de wiskundige formule) in het alfabet verplaatst om het te coderen.

Het betreft hier een type substitutiecipher waarbij elke letter in het alfabet wordt gemapt op een andere letter met behulp van een wiskundige functie. Substitutie betekent eenvoudigweg het vervangen van één ding door een ander. Het basisprincipe van de Affiene versleuteling kan worden uitgelegd door de volgende wiskundige uitdrukking:

[C \equiv aP + b \mod m]

Hierin is:

- (C) de uitkomst (de gecodeerde letter),
- (P) de plaintext letter die je wilt coderen (voorgesteld als een getal waarbij a=0, b=1, ..., z=25),
- (a) en (b) zijn de sleutels gekozen door degene die de code maakt (en ze moeten bepaalde eigenschappen hebben, zoals dat (a) en (m) copriem moeten zijn),
- (m) is de grootte van het alfabet (in het Engels zou (m) 26 zijn: 26 letters),
- "mod" staat voor modulo, een soort rest-na-deling operatie.

Laten we een voorbeeld bekijken om dit concept te verduidelijken:

Stel we kiezen de sleutels (a = 5) en (b = 8), en we willen de letter "h" coderen. Eerst vertalen we de letter "h" naar een getal: h=7 (a=0, b=1, c=2, ..., h=7, ...). Nu vervangen we (P) met 7 in onze formule en rekenen we het uit:

[C \equiv (5 \times 7 + 8) \mod 26] [C \equiv (35 + 8) \mod 26] [C \equiv 43 \mod 26] [C \equiv 17]

Nu vinden we welk karakter overeenkomt met het getal 17: r (a=0, b=1, ..., q=16, r=17, ...). Dus in onze code wordt de letter "h" vervangen door de letter "r".

Om te ontcijferen, moet je de inverse van het getal a vinden (dus een getal dat, wanneer vermenigvuldigd met a, gelijk is aan 1 mod 26). Dit gebruik je dan in de ontcijferformule:

[P \equiv a^{-1}(C - b) \mod m]

- Je bestudeert en zoekt op hoe affiene versleuteling werkt
- Je schrijft een script dat volgens deze versleuteling encrypteert en decrypteert.

Oefening 11: de affiene versleuteling bruteforcen

Jouw opdracht: je gaat een met vermenigvuldigversleuteling geëncrypteerde boodschap decrypteren. Je kent de sleutel niet. Je gaat dus voor elke mogelijke sleutel een decryptie tonen. Je maakt gebruikt van oefening 6 om automatisch de juiste gedecrypteerde boodschap te vinden.

Oefening 12: eenvoudige substitutieversleuteling

Eenvoudige substitutieversleuteling is een type cryptografie waarbij elke letter of teken in een tekst wordt vervangen door een andere, volgens een vooraf bepaalde regel of sleutel. Het basisidee is dat elke letter in de originele tekst (de klare tekst) wordt 'vertaald' naar een nieuwe letter, cijfer, of symbool in de versleutelde tekst.

Hier zijn een paar kernpunten:

- **Sleutel**: De methode of regel die gebruikt wordt om letters te vervangen. Dit zou bijvoorbeeld een verschuiving van enkele posities in het alfabet kunnen zijn (zoals bij het Caesar-cijfer) of een meer complexe regel waarbij elke letter in het alfabet een unieke, willekeurige vervanger krijgt.
- **Vervanging**: Elke letter in de klare tekst wordt vervangen door de corresponderende letter, cijfer, of symbool volgens de sleutel.
- **Omkeerbaarheid**: Het is cruciaal dat de substitutie omkeerbaar is, zodat de ontvanger de originele tekst kan herstellen door de versleutelde tekst te ontcijferen met de juiste sleutel.

Bijvoorbeeld: Stel je hebt de volgende sleutel voor substitutieversleuteling: [\text{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}] [\text{Q W E R T Y U I O P A S D F G H J K L Z X C V B N M}]

Dan wordt de letter "A" in de klare tekst vervangen door "Q" in de versleutelde tekst, "B" wordt vervangen door "W", "C" wordt vervangen door "E", enzovoorts. Als we het woord "CODE" willen versleutelen, dan wordt dat "RWTY" volgens de hierboven gegeven sleutel.

De eenvoudige substitutieversleuteling is vrij eenvoudig te begrijpen maar kan, afhankelijk van de sleutel, variëren in complexiteit en veiligheid tegen ontcijfering.

- Je bestudeert en zoekt op hoe eenvoudige substitutieversleuteling werkt
- Je schrijft een script dat volgens deze versleuteling encrypteert en decrypteert.

Oefening 13: eenvoudige substitutieversleuteling hacken

Deze oefening is niet verplicht: schrijf een script dat een met eenvoudige substitutieversleuteling geëncrypteerde boodschap probeert de decrypteren.