Tinpro01-7 Practicumopdracht 2

W. Oele

27 februari 2016

Inleiding

In deze practicumopgave gaan we ons bezighouden met encryptie. We richten ons daarbij op RSA encryptie, te weten:

- conversie van een tekst naar de ascii waarden van die tekst
- privé en publieke sleutel generatie
- rsa encryptie van een bericht
- rsa decryptie van een bericht

Aan het einde van deze opdracht ben je in staat sleutels te genereren en een gegeven bericht te versleutelen en ontsleutelen. Je schrijft dus een volledig werkende implementatie van dit veelgebruikte algoritme.

Voor deze opdracht heb je een aantal hulpfuncties nodig. Deze functies schrijf je als eerste, zodat je ze verderop in de opdracht kunt gebruiken. Schrijf de volgende functies:

Opdracht 1a

```
euclid::Integer->Integer->Integer
euclid x y
```

Deze functie berekent de grootste gemene deler van twee gegeven natuurlijke getallen x en y. Let op de type signature. Deze werkt met `Integer`, zodat je met grote getallen kunt werken als dat nodig is.

Opdracht 1b

Gegeven de volgende congruentie:

$$e \cdot d = 1 (mod\ m)$$

Een dergelijke berekening is nodig voor het genereren van de publieke en privé sleutel. Een algoritme, waarmee de gegeven congruentie is op te lossen is het volgende:

```
egcd :: Integer -> Integer -> (Integer,Integer,Integer)
egcd 0 b = (b, 0, 1)
egcd a b =
   let (g, s, t) = egcd (b `mod` a) a
    in (g, t - (b `div` a) * s, s)
```

Helaas levert dit algoritme soms een negatieve uitkomst. Gebruikt het algoritme in een eigen versie die bij een negatieve uitkomst de uitkomst positief maakt door er de modulus bij op te tellen.

Opdracht 2: sleutel generatie

Rsa encryptie werkt met twee sleutels:

- de privé sleutel: strikt geheim en in jouw persoonlijke bezit.
- de publieke sleutel: deze mag iedereen hebben.

Voor het genereren van sleutels zijn twee priemgetallen, p en q nodig. Kies twee priemgetallen. Hou deze getallen klein, d.w.z. tussen de 100 en 500. In echte toepassingen van rsa encryptie zijn de getallen veel groter en worden veel efficiëntere algoritmen gebruikt. Deze algoritmen maken gebruik van geavanceerdere getaltheorie. In deze opdracht houden we het bij de basale werking van rsa encryptie.

De volgende berekeningen moeten worden uitgevoerd:

- De modulus: $m = p \cdot q$
- Eulers totient functie: $m' = \phi(m) = (p-1) \cdot (q-1)$

Vervolgens kiezen we een getal e dat relatief priem is met m'. Het getal e voldoet dus aan de volgende twee voorwaarden:

- $e < m'$
- $ggd(e, m') = 1$

Zodra we een geschikt getal e gekozen hebben, moeten we een bijbehorende d berekenen. Voor d geldt:

- $e \cdot d = 1 (mod\ m')$

We hebben nu de privé sleutel, de publieke sleutel en de modulus!

- De privé sleutel is e
- De publieke sleutel is d
- De modulus is m

Opdracht 3a: rsa encryptie

Schrijf een functie die een tuple van sleutel en modulus, alsmede een getal als parameters heeft en dit getal versleutelt:

```
rsaencrypt::(Integer,Integer)->Integer->Integer
rsaencrypt (e,m) x
```

Opdracht 3b: rsa decryptie

Schrijf een functie die een tuple van sleutel en modulus, alsmede een getal als parameters heeft en dit getal ontsleutelt:

```
rsadecrypt::(Integer,Integer)->Integer->Integer
rsadecrypt (e,m) x
```

Opdracht 4

Versleutel en ontsleutel één letter m.b.v. de functies uit opdracht 3. Voor de conversie van een letter naar een ascii waarde en vice versa zijn twee handige functies beschikbaar:

- `ord`
- `chr`

Oefen met beide functies.

Opdracht 5

Alice en Bob willen veilig met elkaar communiceren. Echter: een bericht dat door Alice met haar privé sleutel werd versleuteld, kan door iedereen met de publieke sleutel worden ontsleuteld. Hoe kun je rsa gebruiken om Alice en Bob toch veilig met elkaar te laten communiceren?

Opdracht 6 (facultatief)

Simuleer een man in the middle attack.