Breek de code

Programmeeropdracht 5
Algoritmen en Complexiteit 2022

1 Probleembeschrijving

RSA is een asymmetrisch encryptiealgoritme. Dit houdt in dat er geen geheime sleutel is die zowel voor het encrypten als het decrypten van berichten kan worden gebruikt. In plaats daarvan heb je een public key die je alleen kunt gebruiken voor het versleutelen van berichten en een private key die je alleen kunt gebruiken voor het decoderen van berichten. Nu kun je je public key best aan iedereen meegeven, ze kunnen er namelijk alleen berichten mee versleutelen voor jou en geen berichten van anderen aan jou meelezen (daarvoor is de private key nodig).

Zulke encryptiealgoritmes berusten er op dat sommige wiskundige operaties heel makkelijk zijn de ene kant op, maar veel tijd kosten de andere kant op. Bij het RSA-algoritme wordt gebruikgemaakt van de notie dat het gegeven twee priemgetallen heel makkelijk is om het product van deze twee getallen uit te rekenen, maar dat gegeven alleen het product er geen efficiënte (niet exponentiële) manier bestaat om de bijbehorende priemgetallen te vinden.

Het versleutelen van een bericht via RSA gaat op de volgende manier:

1. Er worden twee priemgetallen p en q gekozen en $n = pq$ wordt berekend.
2. De totiëntfunctie $\phi(n) = (p-1)(q-1)$ wordt berekend en er wordt een getal e gekozen tussen 1 en $\phi(n)$ zodanig dat $\text{ggd}(\phi(n), e) = 1$ (de ggd is de grootst gemene deler).
3. Vervolgens wordt de multiplicatieve inverse d van e modulo $\phi(n)$ berekend. Met andere woorden, er wordt een d gevonden, zodanig dat $de = 1 \mod \phi(n)$.
4. Nu kan een bericht m worden versleuteld met $c = m^e \mod n$ en weer worden gedecodeerd met $m = c^d \mod n$. De public key is dus e en de private key d.

Jij werkt bij de nationale veiligheidsdienst en hebt net een flink aantal met RSA versleutelde berichten onderschept. Je hebt echter ontdekt dat de n die is gebruikt voor deze berichten niet groot genoeg is. Kun jij de berichten ontcijferen?

2 Invoer/uitvoer

2.1 Invoer

De eerste regel bevat het aantal encrypted berichten a die je moet ontcijferen. Vervolgens staan op de volgende a regels respectievelijk het product van de priemgetallen n, de public key e en het versleutelde bericht c.

2.2 Uitvoer

Als uitvoer wordt voor elke regel in de invoer het gedecodeerde bericht m verwacht.

2.3 Voorbeeldinvoer

```
2
854291 5 626144
719501 7 15105
```

2.4 Voorbeelduitvoer

```
656609
481488
```

3 Inleveren

Controleer goed of jouw Pythonversie Python 2 of Python 3 is en zorg dat je Python 3 gebruikt.

Let op: Deze programmeeropdracht wordt voor een groot deel automatisch nagekeken. Zorg dus dat je je precies aan onderstaande richtlijnen houdt, anders kan je code niet worden nagekeken.

Controleer of je code goed werkt voor het inleveren

Door `check.sh` uit te voeren in dezelfde map als je python- en input/output- files kun je controleren of je code de juiste uitvoer geeft op de kleine input (`small.input`) en het juiste inputformaat heeft.

Nakijkeisen

- De code dient geschreven te zijn in **Python 3**. Er wordt nagekeken met Python versie **3.9**.
- De code leest de input van *stdin* en schrijft naar *stdout*. Om eenvoudig invoer te testen kun je gebruikmaken van input redirection:
  `python ex5.py < small.input`

  Hint: de volgende voorbeeldcode leest op de juiste manier van stdin en schrijft hetzelfde naar stdout.

```
import fileinput
for line in fileinput.input():
    # line bevat al een newline van zichzelf
    print(line, end="")
```

- Het codebestand zelf moet de naam `ex5.py` hebben.
- Lever **alleen** het python bestand met je code in en geef het de naam `ex5.py`. Lever dus geen zip/tar in of iets dergelijks.

- Je uitwerking dient binnen 10 seconden antwoord te kunnen geven op zowel de large als small input.

Hoe wordt de code nagekeken?

Er zijn twee inputfiles aan de hand waarvan je code wordt nagekeken:

- Bijgeleverd is `small.input` met de bijbehorende output `small.output`. Als je code dezelfde output geeft met `small.input` als input, wordt in principe het cijfer 6 toegekend.
  NB: Om deze score te behalen is het wel noodzakelijk dat je daadwerkelijk redelijkerwijs het algoritme hebt geïmplementeerd. Simpelweg altijd dezelfde output retourneren wordt niet goedgerekend.
- Daarnaast is een large.input bestand bijgeleverd (zonder de bijbehorende output). Als je code ook de juiste output voor deze input geeft, wordt in principe het cijfer 10 toegekend.
- Op CodeGrade (zie Canvas) kun je direct zien of je code werkt voor small.input. De resultaten voor large.input worden pas na de deadline kenbaar gemaakt.

De code wordt hiernaast gecontroleerd op plagiaat van anderen en code op het internet. Hierop is de algemene plagiaat- en frauderegeling van de UvA van toepassing.

Vragen

Mocht je vragen/problemen hebben over het inputformaat of algemeen over de code, stel een vraag tijdens het werkcollege of stel je vraag op Canvas / via email.