0,3
0.3

AA-Entrega-3

Oriol Miró López-Feliu

March 2023

1 Problema 2

> 2EXP modular. Doneu un algorisme de temps polinòmic que amb entrada els enters a, b, c i un nombre primer p computi $a^{b^c} \mod p$.

ok

El teorema de Fermat estableix que si p és un nombre primer i a és un nombre enter coprimer amb p, llavors $a^{p-1} \equiv 1 \mod p$. Utilitzarem aquesta propietat per tal de simplificar el problema. En particular, calcularem primer $x = b^c \mod (p-1)$, per a poseriorment poder fer simplement $a^x \mod p$

Demostració de l'equivalencia:

$$\begin{aligned} a^{b^c} \mod p &= a^{k(p-1)+x} \mod p \\ &= (a^{p-1})^k \cdot a^x \mod p \\ &= 1^k \cdot a^x \mod p \\ &= a^x \mod p \end{aligned}$$

Calcular $b^c \mod (p-1)$ es pot fer en temps polinòmic, mitjançant l'algorisme d'exponenciació vist a classe amb cost $O(n^3)$, sent n el màxim de les quantitats de bits de b i $p-1$.

Una vegada hem calculat aquest resultat en temps polinòmic, tornem a aplicar el mateix l'algorisme per computar $a^x \mod p$, en temps un altre cop $O(n^3)$.

La descripció sencera de l'algorisme es la següent:

Algoritme 2EXP modular

Entrada: nombres enters a, b, c, p amb p primer i a coprimer amb p.
Sortida: $a^{b^c} \mod p$

1. Calcular $x \leftarrow b^c \mod (p-1)$ mitjançant l'algorisme d'exponenciació ràpida.
2. Calcular $a^x \mod p$ mitjançant l'algorisme d'exponenciació ràpida.

La correcció de l'algorisme es basa en la demostració anterior. La complexitat temporal és polinòmica en funció de la mida de les entrades, amb un cost de $O(n^3)$, sent n el màxim entre les quantitats de bits de b, p i a.

2 Problema 5

> Sistema criptogràfic segur?. Suposem que en lloc d'utilitzar un nombre compost $N = p * q$ com es fa en el sistema rsa, utilitzem un nombre primer p. Per encriptar un missatge $m \mod p$ farem servir un exponent e, de la mateixa manera que es fa en el sistema RSA. L'encriptament del missatge $m \mod p$ seria $m^e \mod p$.
>
> Demostreu que aquest nou sistema no és segur donant un algorisme eficient per desencriptar. Es a dir, doneu un algorisme que, amb entrada $p, e, m^e \mod p$, computi $m^d \mod p$ eficientment. Justifica la correctesa de l'algorisme i analitza el seu temps de computació.

Per tal de poder desencriptar el missatge m donat, hem de calcular $m^d \mod n$, on $d \equiv e^-1 \mod \phi(n)$. Això normalment no es pot fer, degut a que al no conèixer la factorització que produeix n, ens és computacionalment inassolible calcular $\phi(n)$.

En aquest exercici, però, coneixem $n = p$, i al ser p un número primer, és immediat calcular $\phi(n) = p - 1$. Per tant, podem calcular d utilitzant $EXT - EUCLID$, passant com a paràmetres e i $p - 1$. L'algorisme ens donarà com a resultat els valors a, b, c tals que $a * e + b * (p - 1) = c = gcd(e, p - 1)$, i per tant $a \equiv d^-1 \mod (p - 1) = a \equiv e^-1 \mod \phi(p - 1)$, i $a = d$, el valor que estàvem buscant.

Per tant, tenint el missatge encriptat $x = m^e \mod n$, podem desencriptar-lo com $y = x^d \mod n$.

La descripció completa de l'algorisme per desencriptar és la següent:

Algoritme Desencriptar RSA Insegur

Entrada: nombres enters p, e, x amb p primer, e coprimer amb $\phi(p)$ i $x = m^e \mod p$ (missatge encriptat)
Sortida: y tal que $y = x^d \mod n$, on $d \equiv e^-1 \mod \phi(p)$ (missatge desencriptat)
1. Calculem $\phi(p) = p - 1$, ja que p és primer.
2. Calculem $d \equiv e^{-1} \mod \phi(p)$ utilitzant l'algorisme d'Euclides extés.
3. Calculem $y = x^d \mod p$ mitjançant l'algoritme d'exponenciació ràpida per obtenir el missatge original m.

2.1 Correctesa

L'algorisme és correcte perquè si $m^e \mod p$ és el missatge encriptat, llavors $(m^e)^d \equiv m \mod p$ segons el teorema d'Euler, ja que p és primer. Per tant, $(m^e)^d \equiv m \mod p$ implica que $m^d \equiv (m^e)^{-1} \mod p$, i d és el residu de la divisió euclidiana de e^{-1} per $p - 1$, que es pot calcular eficientment amb l'algorisme de l'extensió d'Euclides.

2.2 Cost

EXT-EUCLID té un temps de computació de $O(n^3)$, on n es el màxim entre el nombre de bits de les entrades. El posterior càlcul de $x^d \mod p$ es realitza mitjançant l'algorisme d'exponenciació ràpida, que té cost $O(n^3)$, sent n el màxim entre les quantitats de bits de les entrades.

Al tenir un algoritme on tots els passos requereixen temps polinòmic, l'algoritme té temps polinòmic, concretament $O(n^3)$ on n es el màxim entre la quantitat de bits de e, p i x.