Ros Roger, Alexandre

Març 2023

2. Factorial Modular. Donats dos enters x i N, calcula $x! \pmod N$

1. Demostreu que un enter y és primer si, i només si, per a tot enter $x < y$ es compleix que $\gcd(x!, y) = 1$.
2. Considereu l'apartat previ per demostrar que si `Factorial Modular` fos computable en temps polinòmic, aleshores el problema de `Factoritzar` també seria computable en temps polinòmic.

Solució:

Començarem demostrant el lemma proposat a l'apartat 1 de l'exercici.

Per una banda, tenim que si y és primer llavors $\forall x < y \ \ \gcd(x!, y) = 1$. Sigui $F(x!) = p_0^k p_1^{k'} \dots p_n^{k(n)}$ la factorització de $x!$ en nombres primers i suposem que $\gcd(x!, y) \neq 1$, llavors $\exists\, p \in F(x!)$ s.t. $p|y$. Tot i així, donat que tot $p \in F(x!)$ és menor que y (doncs $p \leq x! < y$), y no seria un nombre primer. Contradicció. $\gcd(x!, y) = 1$.

Per l'altra banda, tenim que si $\forall x < y \ \ \gcd(x!, y) = 1$, y ha de ser primer. Si y no fos primer, llavors $\exists\, p < y$ s.t. $p|y$. Prenent $x = y - 1$, tenim que $p|(y-1)! \implies p|(x!)$ per definició del factorial $(y-1)! = (y-2) \times (y-3) \dots \times 2$ i per l'existència d'aquesta $p \leq y - 1$. Donat que $p|(x!)$, $p|y$ i $x < y$, tenim una contradicció i, per tant, y ha de ser primer. ■

Per a l'apartat 2, suposarem l'existència d'un oracle que, donats els enters x, N calcula $x! \pmod N$ en temps polinòmic respecte x, N.

Donat qualsevol enter $N \geq 2$, el enter positiu més petit α tal que $\gcd(\alpha! \mod N, N) > 1$ és factor primer de N. Això és conseqüència directa del lema demostrat i de l'aplicació del teorema $\gcd(a, b) = \gcd(a \mod b, b)$. Llavors, per a qualsevol $n \geq \alpha$, $\gcd(n! \mod m, m) > 1$.

Això significa que podem emprar una cerca binària per a trobar aquesta α, i serà polinòmic sempre i quan calcular $n! \pmod m$ ho sigui. Aquest algorisme constitueix una reducció de la factorització d'enters al problema del factorial modular.

5. RSA. Demostreu que en comptes de prendre $N = p \times q$ per a la clau pública en RSA prenguéssim N primer, llavors el sistema criptogràfic no seria segur (decriptació fàcil).

Solució:

La seguretat en RSA es basa en la dificultat de factoritzar N. En el cas estàndard on $N = p \times q$ on p, q són dos nombres primers grans, factoritzar N és impossible amb els recursos computacionals actuals. D'aquesta manera, la clau de decriptació només pot ser calculada per a aquells qui coneixen la factorització de N, doncs fàcilment poden computar $\Phi(N) = (p-1) \times (q-1)$.

Si N és primer, però, la factorització ja és coneguda per tothom i tothom pot calcular la clau de decriptació, doncs tothom sap que $\Phi(N) = N - 1$. L'algorisme a emprar per a calcular la clau de decriptació d és l'estàndard.

Donats $N = p$ (mòdul primer), e (clau encriptació pública), $c \equiv m^e \pmod p$ cyphertext, primer calculem d mitjançant Extended Euclid [recordem que $e \times d \equiv \pmod{\Phi(N)}$, per tant cerquem l'invers multiplicatiu $e^{-1} \pmod{\Phi(N)}$] i llavors simplement fem l'exponenciació $c^d \pmod N$, que hauria de donar $m \pmod N$.