PRÁCTICA 1

ARITMÉTICA MODULAR

Esta práctica nos introduce en algunos algoritmos de aritmética modular que son necesarios para el desarrollo de la criptografía de clave pública. Se trata de conocer e implementar algunos de ellos, y de observar las diferencias del tiempo de cálculo entre distintas operaciones. A lo largo de toda la práctica se utilizarán números naturales con un número elevado de cifras, por lo que será necesario optar por un lenguaje de programación que admita dichos cálculos, o bien adaptar, mediante el uso de bibliotecas específicas, el habitual para llevar a cabo estas tareas.

Para el desarrollo de la práctica es conveniente recordar el algoritmo extendido de Euclides, junto con su aplicación para calcular inversos modulares. No obstante, estos algoritmos no son objeto de la práctica.

Lo que hay que realizar es lo siguiente:

En primer lugar, hay que implementar un programa que pida como entrada 3 números naturales, a, b, m, y dé como salida el número a^b mód m.

En segundo lugar, hay que implementar un programa que pida como entrada 3 números naturales, a, b, p y decida si existe $log_a(b)$ mód p, y caso de que exista, debe dar una solución.

La solución puede no ser única. En tal caso, se puede optar por:

- Dar una solución cualquiera.
- Dar la menor solución.
- Dar todas las soluciones menores que p.

Para que el algoritmo funcione bien, el número p debe ser primo. En principio no hay que comprobar si, una vez introducidos los números, el tercer número es primo.

En tercer lugar, hay que realizar un estudio de los tiempos que emplea cada uno de los dos algoritmos anteriores. Para eso, tomamos números primos desde 5 cifras en adelante. Si p es uno de esos números primos, entonces:

- Elegimos al azar dos números a y b, menores que p, pero del mismo tamaño. Estos números podemos tomarlos nosotros. No es necesario utilizar una función que genere números aleatorios.
- Calculamos $c = a^b$ mód p, y medimos el tiempo que tarda.
- Calculamos $log_a(c)$ mód p, y medimos el tiempo que tarda.

Si llegado un número primo para el que estamos haciendo la prueba el algoritmo tarda mucho, no es necesario seguir con el siguiente.

A continuación va una lista de posibles números primos.

$$
\begin{gathered}
46381 \\
768479 \\
9476407 \\
36780481 \\
562390847 \\
1894083629 \\
65398261921 \\
364879542899 \\
8590365927553 \\
28564333765949 \\
123456789101119]
\end{gathered}
$$

Por último, hay que elegir resolver uno de los cuatro problemas siguientes:

1. Factorización de números enteros.
2. Cálculo de raíces cuadradas modulares.
3. Sistemas de congruencias.
4. Curvas elípticas.

Una vez elegido un problema, explicaré exactamente qué es lo que hay que hacer.

De la práctica, hay que entregar:

★ El código fuente.

★ Los ficheros ejecutables (para linux).

★ Un breve manual en el que se expliquen las instrucciones para la utilización del programa.

★ La tabla comparativa de los tiempos empleados en los cálculos de la potencia y el logaritmo.

La fecha límite para la entrega es el día 19 de marzo. Los ficheros hay que subirlos al SWAD, a la zona de ”Mis trabajos”. Si la práctica se hace en grupo, es suficiente con que la suba uno del grupo. Pero hay que indicar quienes son los componentes.

Para que la práctica se evalúe debe ser defendida por todos los miembros del grupo.