# Lab 4: Cryptography

## Pollard’s p-1 algorithm:

## Algorithm description:

The algorithm is a number theoretic integer factorization algorithm, with the idea behind it being to make the exponent a large multiple of p − 1 by making it a number with very many prime factors.

## Basic Algorithm:

The basic algorithm can be written as follows:

**Inputs**: *n*: a composite number

**Output**: a nontrivial factor of *n* or failure

1. select a smoothness bound *B*
2. define {\displaystyle M=\prod \_{{\text{primes}}~q\leq B}q^{\lfloor \log \_{q}{n}\rfloor }} (note: explicitly evaluating *M* may not be necessary)
3. randomly pick *a* coprime to *n* (note: we can actually fix *a*, e.g. if *n* is odd, then we can always select *a* = 2, random selection here is not imperative)
4. compute *g* = gcd(*aM* − 1, *n*) (note: exponentiation can be done modulo *n*)
5. if 1 < *g* < *n* then return *g*
6. if *g* = 1 then select a larger *B* and go to step 2 or return failure
7. if *g* = *n* then select a smaller *B* and go to step 2 or return failure

## Methods:

## modularExp: the method takes as parameters a number, a power and another number and then computes the modulus exponential of the expression

## randomBigInt: returns a BigInteger number with a random byte array (using the random class)

## gcd: returns the greatest common divisor of 2 numbers (bigintegers)

## isPrime: checks the primality of a number (biginteger)

## factorial: returns n!

## factoriseS: computes the factorization of a number by checking iteratively through all numbers smaller than the provided number

## factoriseP1: computes the factorization of a number by utilizing the pollard p-1 method;