Cifrado con RSA

Matemática Discreta

UPV/EHU

Resumen: Se pretende cifrar y descifrar mensajes utilizando el algoritmo RSA con clave pública y privada. Para ello primero hay que codificar el mensaje y a continuación cifrar el mensaje para poder enviarlo de forma segura. Quien reciba el mensaje deberá hacer uso de la clave privada para descifrar y posteriormente descodificar el mensaje.

1. El cifrado mediante el algoritmo RSA

Para cifrar el mensaje codificado hay que hacer uso de los valores n y r de la clave pública: cada código C_i lo vamos a cifrar y convertirlo en R_i mediante la exponenciación modular de la siguiente manera,

$$R_i = C_i^r \bmod n.$$

Los números que se obtengan con ese cifrado son los que van a viajar por el medio de transmisión por el que haremos llegar el mensaje a quien vaya dirigido. En el destino deberán descifrar el mensaje, y para ello deberán hacer uso de la clave privada s y de n que es pública. El descifrado vuelve a utilizar la exponenciacion modular:

$$C_i = R_i^s \bmod n$$

2. Funciones del paquete numbers y de R

Para la cocificación y cifrado del mensaje nos interesan las siguientes funciones de R:

- Cálculo de la exponenciación modular:
  `modpower (a,b,n)` $= a^b \bmod n$

- Cálculo de los códigos ASCII de las letras. Se puede hacer letra a letra o todo el string de golpe
  `strtoi(charToRaw("kaixo"),16L)`
  107 97 105 120 111

- Proceso inverso: de código ASCII a letra o de golpe, de vector de códigos ASCII a string
  `vcodificado<-c(107,97,105,120,111)`
  `rawToChar(as.raw(vcodificado))`
  "kaixo"

No hay que olvidar que los números que tengan más de 12 dígitos se muestran en pantalla mediante la notación científica, por lo que si queremos mestrar todos los dígitos del número debemos anular tal opción:
`format(a, scientific=FALSE)`.

3. Ejercicios

1. Codificar/decodificar

   - Escribir la función codifica(txt) que recibe como parámetro el texto en forma string "txt"y que devuelve el vector de códigos ascii que les corresponde a los caracteres del string.

     ```
     codifica  <- function(txt)
       {
       }
     ```

     Pruebas de que funciona correctamente:

     ```
     texto1<-"kaixo"
     codevector1<-codifica(texto1)
     codevector1
     # 107  97 105 120 111

     texto2<-"KAIXO"
     codevector2<-codifica(texto2)
     codevector2
     # 75 65 73 88 79

     texto3<-"Zer moduz?"
     codevector3<-codifica(texto3)
     codevector3
     ```

```
#  90 101 114  32 109 111 100 117 122  63
```

- Escribir la función decodifica(codetxt) que recibe como parámetro el vector de códigos ASCII y que devuelve el string correspondiente:

```
decodifica <- function(codetxt)
  {
  }
```

Pruebas de que funciona correctamente:

```
txt1<-decodifica(codevector1)
txt1
  # "kaixo"

txt2<-decodifica(codevector2)
txt2
  # "KAIXO"

txt3<-decodifica(codevector3)
txt3
  # "Zer moduz?"
```

2. Cifrar/descifrar

- Escribir la función cifrar(vectorcodigos,r,n) que cifre cada código del vector de códigos que recibe como parámetro. El cifrado se realiza mediante la exponenciación modular: c^r mod n, y tanto r como n son parámetros de la función.

```
cifrar <- function(vectorcodigos,r,n)
  {
  }
```

El resultado es un vector de códigos cifrados, con tantos elementos como elementos hubiera en "vectorcodigos", pero los valores cifrados dependen de los valores r y n que se hayan utilizado. Un ejemplo con los valores de $n = 9797$ y $r = 7$ son:

```
vectorcifrado1<-cifrar(codevector1,7,9797)
vectorcifrado1
# 2792 5432 4668 4973 7969

vectorcifrado2<-cifrar(codevector2,7,9797)
```

```
vectorcifrado2
# 7976 4764 2565 8540 4974

vectorcifrado3<-cifrar(codevector3,7,9797)
vectorcifrado3
#  375 2222 7721 3675  493 7969 6261 8564 4122 4604
```

- Escribir la función descifrar(vectorcifrado,r,n) que descifre cada código cifrado del vector que recibe como parámetro. El descifrado se realiza mediante la exponenciación modular: m^s mod n, y tanto s como n son parámetros de la función. En este momento habría que preguntarse si realmente nos hace falta esta función.

  ```
  descifra <- function(vectorcifrado,s,n)
    {
    }
  ```

  El resultado debería ser el vector de códigos ASCII original del mensaje, pero para ello debemos utilizar la clave privada adecuada a la clave pública que se haya utilizado para cifrar el mensaje. En los ejemplos de cifrado se han utilizado $n = 9797$ y $r = 7$ y la clave privada que les corresponde es $s == 2743$. Ante la pregunta de si hacía falta la función de descifrar, comparemos los resultados de descifrar con el resultado de volver a cifrar los mensajes cifrados, pero con el parámetro s sustituyendo a r. Veamos que pasa:

  ```
  descifrado1<-descifra(vectorcifrado1,2743,9797)
  descifrado1
  # 107  97 105 120 111
  descifrado11<-cifrar(vectorcifrado1,2743,9797)
  descifrado11
  # 107  97 105 120 111 (?)
  descifrado2<-descifra(vectorcifrado2,2743,9797)
  descifrado2
  # 75 65 73 88 79
  descifrado21<-cifrar(vectorcifrado2,2743,9797)
  descifrado21
  # 75 65 73 88 79
  descifrado3<-descifra(vectorcifrado3,2743,9797)
  descifrado3
  #  90 101 114  32 109 111 100 117 122  63
  descifrado31<-cifrar(vectorcifrado3,2743,9797)
  descifrado31
  ```

```
#  90 101 114  32 109 111 100 117 122  63
```

vemos que descifrar y volver a cifrar con s tiene el mismo resultado. Se puede probar con diferentes claves RSA y ver los resultados.

3. **El juego.** Vamos a intercambiar mensajes entre nosotros. Para ellos seguiremos los siguientes pasos:

   a) Primero debemos generar nuestras claves. Cada cual sus claves basadas en dos numero primos con valores entre 200 y 500 (para que no nos salgan números demasiado grandes).

   b) Una vez que tengamos las claves RSA, publicaremos en la pizarra (o en el chat) nuestras claves públicass, con lo que cualquiera podrá enviarnos un mensaje cifrado.

   c) Elegiremos una persona a la que vayamos a enviar un mensaje, y copiaremos su clave pública. Con dicha clave cifraremos el mensaje y escribiremos en la pizarra o en el chat el mensaje cifrado.

   d) La persona que ha recibido el mensaje en la pizarra deberá descifrar con su clave privada y escribir el mensaje descifrado al lado.

   e) Pero cualquiera puede calcular la clave privada a partir de la pública, por lo que podemos interceptar los mensajes incluso antes de que sean descifrados en desino. Si lo hacemos podemos escribir el mensaje en la pizarra antes que quien debería haberlo escrito...

   Este juego en la pizarra se puede realizar mediante el correo electrónico o chat y con números de muchos dígitos, para dificultar el trabajo a quienes quieran espiarnos.