ITAM

Sistemas Distribuidos

Salvador García González - 119718
Maestría en Ciencias en Computación

Tarea Extra 1

Ejercicios Criptografía

Primavera 2021

Profesor:
Marcelo Mejía Olvera

Capítulo 1

Ejercicios extra criptografía

1.1. Ejercicio 1:

```
# En este ejercicio se intenta por fuerza bruta encontrar tanto x como y:

x = 1
break_p <- FALSE

while(!break_p){
  if(3^x %% 47 == 28){
    break_p <- TRUE
    print(x)
  }else{
    x <- x + 1
  }
}

y = 1
break_p <- FALSE

while(!break_p){
  if(3^y %% 47 == 17){
    break_p <- TRUE
    print(y)
  }else{
    y <- y + 1
  }
}
```

De esta forma, x y y son 8 y 10 respectivamente

```
> x
[1] 8
> y
[1] 10
```

1.2. Ejercicio 2:

```r
# Esta libreria nos permite transformar un numero entero a su equivalente binario
library(binaryLogic)

# La function 'construir_vector_base' permite calcular los residuos de base^1, base^2, base^4, base^8 y base^16 o hasta el binario necesario modulo 'mod'.

construir_vector_base <- function(bin, mod, base){
  residuo <- c()

  # Inicializamos el vector residuo con base^1 modulo mod
  inicial <- base^1 %% mod
  residuo <- c(inicial)

  # Llenamos el vector de residuos con los demas residuos de base^i. Para esto utilizamos los elementos anteriores de residuo para evitar computo extra
  for(i in 2:length(bin)){
    num_ant <- (residuo[length(residuo)])
    residuo <- c(residuo, (num_ant*num_ant) %% mod)
  }

  # regresamos el vector residuo
  return(residuo)
}


# obtener residuo nos ayuda a transformar el numero deseado a binario, calcular su vector de residuos y calcular el modulo del numero original
obtener_residuo <- function(exp, mod, base){

  bin <- rev(as.binary(exp))
  residuos <- construir_vector_base(bin, mod, base)
  vec_nums <- as.numeric(as.character(bin)) * residuos
  prod(vec_nums[vec_nums>0]) %% mod

}

# ejemplo (resultado = 12):
obtener_residuo(29, 35, 17)
```

```
> bin
[1] 1 0 1 1 1
> residuos
[1] 17  9 11 16 11
> vec_nums
[1] 17  0 11 16 11
> prod(vec_nums[vec_nums>0])
[1] 32912
> obtener_residuo(29, 35, 17)
[1] 12
```